VOSS

MICRO APPLICATION

19

AMSTRAD

PROGRAMMES ET APPLICATIONS ÉDUCATIFS SUR CPC



UN LIVRE DATA BECKER

VOSS

MICRO APPLICATION

19

AMSTRAD

PROGRAMMES ET APPLICATIONS ÉDUCATIFS SUR CPC



UN LIVRE DATA BECKER

Distribué par : MICRO APPLICATION
13, Rue Sainte Cécile
75009 PARIS

et

EDITION RADIO
9, Rue Jacob
75006 PARIS

ISBN : 2-86899-048-7

Collection dirigée par Mr Philippe OLIVIER

AVERTISSEMENT

On parle beaucoup de l'ordinateur à l'école et l'installation des premières machines a décidé MICRO APPLICATION à publier une collection à vocation éducative.
La puissance et la facilité d'utilisation du BASIC, les excellentes performances techniques du CPC nous ont conduit tout naturellement à adopter ce matériel pour ce premier ouvrage.

Dans ce livre on s'intéresse au traitement des données et à la programmation. Il est facile à comprendre et traite beaucoup de sujets. Mais attention, il n'est pas du tout réservé aux cours ni aux séminaires. Les mordus de l'informatique pourront aborder des sujets qui ne les passionnaient pas outre mesure et les étudiants pourront s'initier à la programmation.

Il serait étonnant que à la maison ou à l'école, l'on ne puisse pas utiliser le CPC d'une manière profitable. Amusez-vous bien !

TABLE DES MATIERES

INTRODUCTION

Un ordinateur, tel que l'AMSTRAD CPC, n'est pas uniquement destiné au jeu. Aussi fascinante que puissent être l'utilisation des jeux d'arcade, d'aventures ou de simulations en tous genres, il est bien plus passionnant de créer ses propres programmes : celui qui, pour la première fois, réalise un programme qui peut résoudre en quelques secondes et sans faute un problème donné, a toutes les raisons d'en être fier. Un ordinateur est capable d'accomplir toutes sortes de calculs et de traitements et peut donc nous décharger d'un travail intellectuel pénible. (ceci explique en partie le succès fulgurant de l'informatique).

Pour toutes ces raisons nous nous proposons dans ce livre de montrer comment on peut se servir d' un ordinateur, et plus particulièrement de l'AMSTRAD CPC, pour résoudre des problèmes importants qui se posent dans la vie de tous les jours, et, entre autres, à l'école. Par conséquent, quatre buts seront poursuivis dans ce livre :

1) On choisira des exercices classiques tels qu'ils sont posés en classe dans les domaines des mathématiques, de la physique des langues, etc. et l'on étudiera les différentes manières de les résoudre.

2) On montrera comment aborder d'une manière générale des problèmes posés pour permettre à l'ordinateur de les résoudre.

3) Le lecteur pourra observer la façon dont est utilisé le BASIC (langage de programmation) pour écrire les programmes qui nous permettrons de résoudre ces problèmes.

4) Le lecteur aura donc à sa disposition, après avoir lu ce livre un échantillon de programmes qu'il pourra utiliser dans la vie de tous les jours ou bien à l'école.

Il va de soi que l'utilisation de ce livre ne sera bénéfique que si le lecteur expérimente lui-même les programmes présentés et s'il en crée d'autres. Le proverbe "c'est en forgeant que l'on devient forgeron" s'applique aussi à la programmation.

On étudiera en détail, au long des chapitres de ce livre, les problèmes qui peuvent être posés dans les domaines suivants :

1 - les mathématiques
2 - la chimie
3 - la physique
4 - les langues
5 - la biologie et l'écologie
6 - l'histoire et la géographie
7 - l'économie
8 - autres domaines

A la suite de ces chapitres deux autres points seront traités :

- **la mise en marche de l'ordinateur**
- **les principales instructions du BASIC**

Le lecteur ayant déjà les connaissances suffisantes pourra sans crainte ignorer ces paragraphes.

Les programmes qui seront développés dans le présent ouvrage sont de deux sortes :

Les programmes devant résoudre des problèmes déterminés : par exemple, un programme ayant pour but de définir le "plus grand commun diviseur". Ces programmes peuvent être "alimentés" par différentes données, chacun d'eux ayant une fonction précise.

Les programmes d'entraînement pur : par exemple, un programme permettant l'apprentissage du vocabulaire anglais ne résoudrait pas de problèmes en tant que tels, mais servirait d'apprentissage.

A propos de ces programmes, il est à noter qu'en munissant l'ordinateur de différents périphériques, leur utilisation sera plus performante.

D'autre part, une cassette et une disquette comportant les programmes dont il sera question dans ce livre permettront au lecteur désireux de les utiliser, de se libérer de la charge que représente leur saisie sur un clavier. Ne sont toutefois pas inclus sur la cassette et la disquette ceux, donnés à titre d'exemples, dans le chapitre qui suit. Ceux-ci n'ayant pour but que d'illustrer les principales instructions du BASIC.

La remarque qui suit est particulièrement importante pour l'utilisation des autres programmes :

Dans les programmes devant résoudre des problèmes donnés à l'école, nous avons intentionnellement renoncé à travailler en BASIC avancé ou avec des instructions trop performantes. Nous nous sommes efforcés d'employer des instructions très simples, lors de l'élaboration des programmes, pour ne pas assaillir le lecteur débutant de questions techniques de programmation, mais au contraire, pour lui permettre d'aller à l'essentiel.

Cette restriction implique que certains programmes peuvent paraître peu élaborés, mais ils fonctionnent malgré tout très bien avec les moyens les plus simples, ce qui était notre but.

Pour résoudre les problèmes plus complexes proposés dans les prochains chapitres, il sera tout de même nécessaire de connaitre de nouvelles instructions BASIC qui seront définies au fur et à mesure.

Nous attirons d'autre part l'attention du lecteur sur le fait que certains programmes présentés ne sont donnés qu'à titre d'exemples.

Si un programme nécessite l'entrée de données, le programmeur, s'il veut utiliser ce programme pour une application précise, pourra allonger la liste des données de l'exemple. Ceci, bien sûr, ne change en rien la structure du programme.

INSTRUCTIONS DE BASE DU BASIC



1.1 Avertissement

Nous nous proposons dans ce chapitre de présenter quelques notions de base propres à l'informatique. Nous retrouverons ces notions par la suite. Les principales instructions du BASIC seront expliquées brièvement.

Le lecteur qui possède des connaissances suffisantes en BASIC peut se permettre de ne pas en tenir compte.

Celui qui, par contre, n'a aucune expérience en matière de programmation trouvera un peu succintes les informations que nous donnons dans ce chapitre. Il découvrira cependant quelles sont les caractéristiques du BASIC en lisant plus avant cet ouvrage, son emploi étant constament impliqué dans notre étude. Ce chapitre joue donc le rôle d'introduction.

1.2 Principes fondamentaux

Le mot "Computer", traduit littéralement, signifie "Calculateur". Aujourd'hui l'ordinateur est capable de faire bien plus que de compter. Il peut par exemple traiter les lettres de l'alphabet, des paroles ou des textes, il est capable de composer de la musique, il peut aussi bien résoudre des problèmes opérationnels que fonctionnels, etc.. Cela signifie que l'ordinateur, contrairement aux grosses unités des années 60, met ses capacités de calcul à la portée de chacun. De plus son prix ést devenu relativement abordable.

Données :

Par "données" , on entend "informations" au sens le plus large. Les données peuvent être, par exemple :

- des chiffres ou des nombres
- des valeurs
- des lettres
- des symboles (caractères spéciaux)
- des mots ou des textes.

Traitement des données

Cette expression recouvre toutes les fonctions (souvent mathématiques) permettant de procéder à la saisie, au stockage, à l'interprétation ou à l'analyse de données, et d'obtenir des résultats exploitables.

Si on effectue le traitement des données tel que nous venons de le définir, il est utile de connaître le vocabulaire suivant :

BIT

Un BIT (Binary Information Digit) est la plus petite unité d'information (donnée) traitée par un ordinateur.
On l'appelle ainsi car il applique formellement le langage binaire. En d'autres termes il ne disposera que de deux possibilités pour stocker une information, qui sont désignées par les chiffres 0 et 1. Un technicien vous l'expliquerait en vous disant que le courant passe ou ne passe pas.

OCTET

Un multiplet correspond à l'assemblage d'une suite de bits.
On considère que 8 bits forment un octet. Un octet est donc capable de stocker une série de 8 zéros et/ou de uns.

SYMBOLE

Nous distinguons trois groupes de symboles :

1 - les symboles numériques = chiffres
2 - les symboles alphabétiques = lettres
3 - les autres symboles = caractères spéciaux

En général, chacun de ces symboles est représenté dans l'ordinateur sous forme de 8 zéros et/ou de uns, ce qui veut dire qu'un symbole est représenté par un seul octet (voir plus haut).

ZONE

Une suite de symboles (des chiffres, des lettres,...) constitue une zone. prenons un exemple : Le mot PRINT est composé de 5 symboles, et utilise 5 octets dans l'ordinateur ; ces 5 symboles associés constituent ce qu'on appelle une zone.

Il en va de même du nombre 178. Il sera stocké dans une zone de trois chiffres (il peut bien sûr exister des zones d'un seul chiffre).

VARIABLE :

Une variable est une quantité non fixe ou plus exactement un nom auquel on peut attribuer différentes valeurs (voir plus bas).

Par exemple, à la variable "taille" peuvent être attribuées les valeurs 178, 185, 167 etc. , à la variable "nombre d'enfants", les valeurs 1, 2, 0, 4, etc. , à la variable "lieu", les valeurs "Paris", "Marseille", "Lyon", etc ...

VALEUR :

Le lecteur déduira facilement de ce qui précède que la notion de "valeur" représente le contenu d'une variable donnée. Une valeur n'est pas nécessairement un nombre, mais, comme le montre l'exemple de la variable "lieu", elle peut être un mot ou un texte.

Les valeurs couvrent toujours une zone de l'ordinateur (voir plus haut). On parle encore d'emplacement mémoire.

TABLEAU :

On appelle "tableau", un ensemble de variables de même nom qui sont différenciées par leur numéro d'ordre dans le tableau qui les regroupe (ce peut être un tableau de nombres ou de mots).

STRING :

On appelle "string" un texte qui peut apparaître comme la valeur d'une variable (on parle en général d'une chaîne de caractères)

ENSEMBLE DE DONNEES :

On appelle ensemble de données plusieurs zones assemblées.
Par exemple, tous les renseignements disponibles sur un individu (nom, sexe, âge, domicile, etc.) constituent un ensemble de données. Ici, il serait composé d'une série de nombres et/ou de mots qui porterait sur un même individu.

Même une séquence qui ne serait composée que de mots peut être considérée comme étant un ensemble de données, par exemple dans cette instruction de programme :

```
PRINT "ANNE","PATRICK","LAURENT"
```

ANNE, PATRICK et LAURENT sont des données.

On désigne encore cet ensemble de données sous le nom d'"enregistrement".

FICHIER :

L'association de plusieurs ensembles de données relatifs forment un fichier. Nous pouvons obtenir un fichier en groupant, par exemple, les ensembles de données relatifs aux employés d'une entreprise. De la même manière, nous obtiendrions un autre fichier si nous considérions toutes les instructions d'un programme.

On appelle encore ce fichier "file".

1.3 Fonctionnement de l'ordinateur

Pour procéder au traitement des données telles que les avons définies précédemment, nous devons exécuter les tâches (ou fonctions) suivantes :

1 - enregistrer les informations (saisie des données)
2 - stocker les informations
3 - traiter les informations au sens littéral (par exemple calcul ou réorganisation des données etc.)
4 - afficher des résultats

La saisie, le traitement et la sortie des données nécessitent une série d'opérations que nous n'expliquerons pas ici, l'ordinateur les exécutant, en général, de lui-même (c'est le système d'exploitation qui s'en charge).

L'ordinateur doit donc recevoir des informations pour que la procédure de traitement des données puisse avoir lieu. A lui ensuite de restituer les informations sous forme de résultats.

Nous devons distinguer trois catégories d'informations à fournir à l'ordinateur :

1 - Les données à traiter
2 - Les instructions de programme (les étapes de la procédure nécessaires pour soumettre les données entrées aux procédures d'évaluation souhaitées).
3 - Les commandes (informations et instructions destinées au système d'exploitation de l'ordinateur).

L'ordinateur se compose de différents éléments qui lui sont nécessaires pour accomplir ses tâches diverses :

1 - Unités d'entrée (par exemple le clavier)
2 - Domaine de traitement (l'"unité centrale")
3 - Unités de sortie (écran ou imprimante par exemple).

Il existe aussi, des mémoires externes (ou périphériques). Elles servent à conserver les programmes ou à stocker les données. En règle générale, il s'agit de cassettes (bandes magnétiques) ou de disquettes qui peuvent être enregistrées ou lues par des mécanismes d'entraînement apropriés.

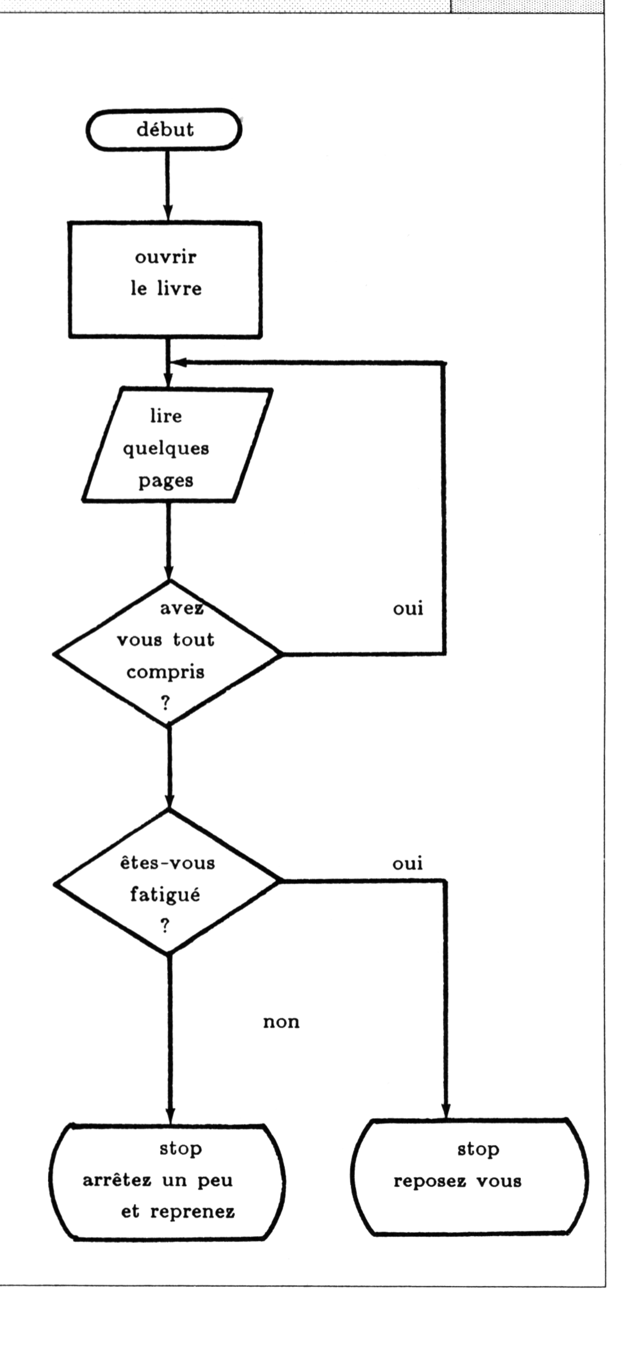
1.4 Analyse du problème

Si un problème donné doit être résolu à l'aide d'un ordinateur, on ne pourra y arriver que si l'utilisateur a déjà trouvé la manière dont il doit s'y prendre (ne jamais oublier cette remarque !). Cela signifie que l'ordinateur est au service de la réflexion humaine. L'ordinateur ne fait qu'exécuter les instructions menant à la solution attendue, nous déchargeant ainsi de cette tâche.

Pour qu'un ordinateur puisse effectuer ce travail, il est nécessaire de lui indiquer la marche à suivre en lui donnant des instructions successives. Pour ce faire le problème à résoudre doit être décomposé (du point de vue des idées) en une suite logique d'étapes distinctes. Nous appelons ce processus : analyse du problème.

Une telle analyse se trouve souvent facilitée si l'on représente graphiquement ces étapes sous forme d'un organigramme.

L'exemple qui suit le montre concrètement :


début
ouvrir
le livre
lire
quelques
pages
avez
vous tout
compris
?
oui
êtes-vous
fatigué
?
oui
non
stop
arrêtez un peu
et reprenez
stop
reposez vous

Nous reviendrons dans les prochains chapitres sur "l'analyse du problème" car elle facilite sensiblement la programmation et permet d'éviter un bon nombre d'erreurs de logique.

1.5 Sortie du résultat

On appelle programme une suite d'instructions qui, exécutées par l'ordinateur permet de résoudre un problème donné. Il est nécessaire d'écrire dans un langage que l'ordinateur puisse comprendre. Le BASIC fait partie de ces langages.

BASIC : **B**eginner's
All-purpose
Symbolic
Instruction
Code

Tout programme en BASIC se compose d'une suite d'instructions ("statements") qui sont introduites successivement dans l'ordinateur. Chaque ligne prend le nom d'enregistrement.

Règle 1 : Toute ligne requiert un numéro d'enregistrement.

Règle 2 : Toute ligne doit être validé par la touche ENTER.

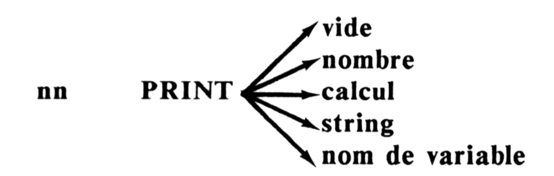
Règle 3 : Une même ligne peut comprendre plusieurs instructions ; celles-ci sont alors séparées par deux points (:).

Instruction 1 :

nn END

L'instruction END indique la fin d'un programme BASIC.

Instruction 2 :



L'instruction PRINT permet d'afficher des résultats à l'écran. On distingue 5 possibilités :

1 - vide : signifie que la ligne est vide

2 - nombre : écriture du nombre

3 - calcul : écriture du résultat

4 - string : écriture du "string" (un "string" est une chaîne de caractères que l'on met entre guillemets)

5 - nom de variable : écriture du contenu de la zone correspondant au nom de la variable

Il est possible de remplacer le mot PRINT par un point d'interrogation (?).

Application de ces possibilités dans le programme BASIC suivant :

```
10 PRINT 3
20 PRINT
30 PRINT 4.8/2
40 ?
50 ?"ANNE"
60 ?
70 ? X
80 END
```

Un tel programme ne peut être exploité qu'après avoir entré la la commande RUN :

Commande 1 :

RUN

Après avoir entré la commande RUN à l'écran, nous obtenons, pour l'exemple donné ci-dessus, les résultats suivants :

3

2.4

ANNE

0

Règle 4 : Les commandes n'ont pas de numéro d'enregistrement. Pour réafficher le programme à l'écran, nous entrons la commande suivante :

Commande 2 :

LIST

Quelques compléments d'information sur l'ordre PRINT :

Les différentes possibilités de l'instruction 2 peuvent être employées plusieurs fois ou être inclues dans un seul ordre PRINT.

On les sépare alors par des virgules ou des points-virgules.

Règle 5 : Si les différentes expressions de l'instruction PRINT sont séparées par des virgules, la visualisation de chacune de ces expressions se fera au début de la prochaine zone d'écran. Si l'on utilise le point-virgule, elle se fera immédiatement après la précédente.

Règle 6 : Tout ordre PRINT provoque, bien sûr, le passage à la ligne suivante. Si l'instruction ne comporte que le mot PRINT, il ne se produit qu'un saut de ligne.

Règle 7 : Une virgule ou un point-virgule à la fin d'une instruction PRINT ne renvoie pas à la ligne.

L'instruction PRINT fait apparaître les résultats à l'extrême gauche de l'écran, il suffit d'employer la fonction TAB (TAB = tabulation).

PRINT TAB (10) 8

renvoie l'expression de la valeur 8, dix espaces sur la droite.

Etudions maintenant les particularités des calculs à réaliser, ainsi que de la notion de variable qui intervenait déjà dans l'instruction PRINT.

Pour effacer un ancien programme on utilise la commande NEW qui efface du contenu de la mémoire le programme :

Commande 3 :

NEW

1.6 Affectation de valeur

Il est en général très utile, nous le verrons par la suite, de donner des noms aux différents emplacements mémoire de l'ordinateur pour pouvoir y stocker ensuite différentes valeurs.

Il suffit de distinguer trois types de variables :

1 - les variables réelles
2 - les variables entières
3 - les variables string

Les variables réelles peuvent, en tant que telles, admettre des nombres réels tels que : 3.5, 17.01, 18., 0.04, -7.4, etc...
(en programmation,on utilise un point à la place de la virgule décimale).

Les variables entières ne peuvent admettre que les nombres entiers, tels que : 7, 8, -3, 0, 20, etc...

Les variables string n'admettent que les suites de symboles qui doivent alors être mises entre guillemets, tels que : "PIERRE", "00", "PARIS", "A-15", etc...

Règle 8 : Les noms des variables réelles se composent d'un ou de deux caractères alphanumériques, le premier devant toujours être une lettre. On ajoute à ce caractère le signe % et le signe $ pour les variables string.

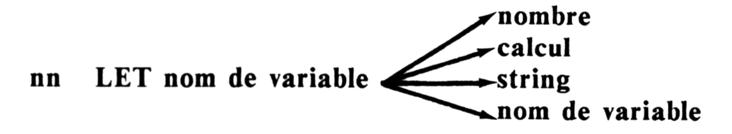
La plupart des ordinateurs admettent des noms plus longs (consulter les manuels d'utilisation).

Exemples :

1 - Variables réelles :	**X**	**B1**	**NN**
2 - Variables entières :	**C%**	**Y1%**	**AA%**
3 - Variables string :	**F$**	**C2$**	**ZZ$**

Sachant cela, nous allons étudier l'instruction LET qui est employée, pour l'affectation d'une valeur, de la manière suivante :

Instruction 3 :



Il est possible de supprimer le mot LET.

Il est également possible d'effectuer des calculs grâce à cette instruction comme avec l'instruction PRINT.

Les opérateurs de calcul sont les suivants :

+	Addition
-	Soustraction
*	Multiplication
/	Division
^	Elévation à une puissance

Le priorité donnée aux opérateurs correspond à ceux qui sont appris en mathématiques à l'école ; On peut forcer ces priorités en mettant des parenthèses.

L'ordre LET permet donc d'écrire des instructions assez complexes. Si l'on fait une faute de frappe, la correction la plus simple consiste à s'arrêter, taper ENTER puis réécrire la ligne en question. S'il faut effacer une ligne complète, il suffit de retaper son numéro suivi de ENTER.

Pour insérer une ligne, lui donner un numéro intermédiaire.

De nombreux calculs sont facilités par ce que l'on appelle des fonctions intégrées dont la syntaxe est la suivante :

nom de variable = nom de fonction (argument)

FONCTIONS IMPORTANTES

NOM	FONCTION
SIN COS TAN	calcule les valeurs fonctionnelles d'angle l'argument est ici exprimé en radian
LOG EXP	logarithme népérien son inverse
SQR	racine carrée
ABS	valeur absolue
INT	partie entière
RND	production de nombres aléatoires (réels compris entre 0 et 1).
CHR$	sortie du code ASCII d'un caractère.

Il est inutile de donner ici des exemples de ces fonctions. Vous en trouverez suffisamment dans les chapitres qui suivent.

1.7 Saisie des données

L'instruction suivante convient remarquablement bien pour la saisie de données :

Instruction 4 :

INPUT ("TEXTE";) liste de variables

Cette instruction, n'est pas aussi compliquée qu'elle peut le paraître :

Lorsque l'ordinateur arrive à cette instruction, il interrompt la procédure du programme et fait apparaître un point d'interrogation sur l'écran (en plus du "texte"). Ce texte est facultatif, c'est la raison pour laquelle il figure entre parenthèses dans l'instruction présentée plus haut. Il attend alors que l'utilisateur lui donne autant de valeurs (séparées les unes des autres par une virgule) qu'il y a de noms de variables dans la liste (soit, au minimum, une valeur s'il n'y a qu'une variable).

EXEMPLES :

INPUT X

L'ordinateur attend un nombre qui sera ensuite stocké en mémoire dans la zone X.

INPUT A,B,C

L'ordinateur attend trois nombres.

INPUT"DONNER LES NOMS S'IL VOUS PLAIT;"N$

L'ordinateur imprime la chaine de caractères qui se trouve entre guillemets, PUIS attend une chaîne de caractères (string) : par exemple "PIERRE" qui sera stockée dans la zone N$.

Règle 9 : Pour l'instruction INPUT, les variables qui figurent dans la liste doivent être séparées par des virgules. Il en va de même des valeurs à introduire.

EXEMPLE :

```
10 INPUT"4 VALEURS S.V.P. : ";A,B,C,D
20 LET S = A + B + C + D
30 LET AM = S/4
40 PRINT "VALEUR MOYENNE = ";AM
50 END
```

Comme on le voit, ce programme calcule la moyenne de quatre nombres quelconques, entrés par l'utilisateur.

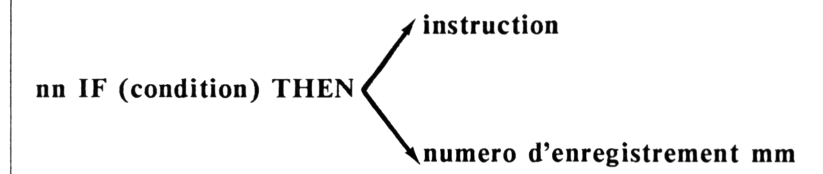
1.8 Branchements

Tant qu'aucun branchement (saut) n'apparait pas dans un programme BASIC, son déroulement se fait dans l'ordre croissant de ses numéros d'enregistrement. Pour changer cet ordre, il est nécessaire d'effectuer des sauts dans le programme.

Nous distinguons deux types de branchements : conditionnels et inconditionnels.

Nous allons maintenant étudier l'instruction qui permet de réaliser un branchement conditionnel :

Instruction 6



Cette instruction conduit le système d'exploitation de l'ordinateur à faire un test : si la condition qui suit le mot IF est remplie, ce qui se trouve derrière le mot THEN sera exécuté, ou on se branchera à l'instruction portant le numéro mm.

Par contre, si la condition n'est pas remplie, on passera à l'instruction qui suit celle du IF.

Règle 10 : Si la condition qui suit le mot IF est remplie, il y aura, selon ce qui suit le mot THEN, exécution d'instructions ou alors branchement à la ligne portant le numéro mm.
Si la condition n'est pas remplie le système passera à l'enregistrement suivant.

Règle 11 : Si, dans le même enregistrement, plusieurs instructions suivent celle du IF, elles ne pourront être exécutées que si la condition du IF est satisfaite.

EXEMPLE :

```
10  I = 1
20  Q = I * I
30  W = SQR (I)
40  PRINT I, Q, W
50  I = I + 1
60  IF I <=20 THEN 20
70  END
```

Comme on le voit, ce programme calcule, pour chaque entier de 1 à 20 (contenu dans la variable I), son carré (contenu dans la variable Q, ligne 20) et sa racine carrée (variable W, ligne 30) puis affiche à l'écran, ces trois valeurs sur une même ligne.

A ce propos, la ligne 60 est extrêmement importante. Elle signifie qu'il faut reprendre le programme à partir de la ligne 20 aussi longtemps que se trouve, dans la variable I et après incrémentation de 1 (ligne 50), une valeur inférieure à 21.

Ainsi, l'ordre de branchement vers la ligne 20 qui figure derrière THEN sera exécuté tant que cette condition sera remplie ; si elle ne l'est plus, c'est l'enregistrement qui suit l'instruction IF (ligne 70) qui sera traité.

L'alternative au saut conditionnel est le saut inconditionnel.

Instruction 6 :

nn GOTO mm

Quand, à l'exécution du programme, le système arrive à cette instruction, un branchement s'effectue à la ligne portant le numéro mm.

```
10    CLS
20    INPUT"ENTRER UNE VALEUR :";X
30    N = N + 1
40    S = S + X
50    INPUT"UNE AUTRE VALEUR (O/N)";A$
60    IF A$ = "O" THEN 20
70    AM = S/N
80    PRINT "MOYENNE = ";AM
90    END
```

Ce programme détermine la moyenne arithmétique d'un nombre quelconque de valeurs saisies. Après chaque entrée de valeur, l'ordinateur demande à l'utilisateur s'il veut en entrer une autre. Si celui-ci répond par "O" (pour "Oui") un branchement se produit de la ligne 60 à 20 où on lui réclame à nouveau une valeur.

Si en revanche, il n'y a plus de valeur à entrer, si donc l'utilisateur ne répond pas par "O" à la ligne 50 (instruction INPUT), l'exploitation du programme reprend à partir de la ligne 70 (calcul et sortie de la moyenne).

1.9 Boucles de programme

Il est souvent utile d'exécuter certaines parties d'un programme plusieurs fois. C'est dans ce but que l'on se sert de ce que l'on appelle des boucles (loops). Il est aisé d'en fabriquer à partir des instructions que nous avons étudiées jusqu'à présent, comme le montre le programme précédent.

La programmation de boucles est facilités par l'utilisation des instructions suivantes qui offrent aussi de nouvelles possibilités.

Instruction 7 :

nn FOR nom de variable = valeur initiale
TO valeur finale (STEP valeur du pas)

Instruction 8 :

nn NEXT nom de variable de travail

La variable de travail qui est apparue dans ces deux instructions doit nécessairement être une variable réelle. La valeur initiale, la valeur finale et le pas de progression peuvent être des nombres, des variables ou des expressions arithmétiques.

Lorsque le système arrive à l'instruction FOR, il donne à la variable de travail la valeur initiale et il exécute les instructions comprises entre FOR et NEXT en tenant compte de cette valeur.

Arrivé à NEXT, il incrémente la variable de travail de la valeur du pas et réexécute le bloc d'instructions compris entre FOR et NEXT en tenant compte cette fois de la nouvelle valeur de la variable de travail.

La boucle est exécutée tant que la valeur de la variable de travail n'a pas dépassé la valeur finale. Dès que le cas se présente on traite la ligne qui suit.

Le pas STEP peut être supprimé de la commande FOR s'il est égal à un.

Le programme suivant imprime 10 fois le mot "BONJOUR", l'un en dessous de l'autre.

```
10 FOR I = 1 TO 10
20 ? "BONJOUR"
30 NEXT I
40 ?
50 ? "FIN"
60 END
```

1.10 Utilisation de la mémoire externe

Il est possible de stocker des programmes pour les réutiliser en cas de besoin, ou d'en utiliser de préétablis grâce aux cassettes ou aux disquettes. Pour les utiliser il faut connaitre les commandes suivantes :

Pour charger un programme d'une cassette :

Commande 4 :

LOAD "Nom du programme"

Après l'exécution de cet ordre par l'ordinateur, le magnétophone est utilisable. Il s'agit alors de veiller à ce que la cassette introduite ne soit pas trop embobinée de façon à ce que l'on puisse retrouver le début de notre programme.

Pour stocker un programme sur une cassette nous utilisons :

Commande 5 :

SAVE "Nom du programme"

Si l'on veut connaître le contenu de la cassette, ou, en d'autres termes, si l'on veut obtenir une table des matières, on tape :

Commande 6 :

CAT

Après avoir suivi les instructions de l'ordinateur, l'utilisateur voit apparaître à l'écran la table des matières de la cassette.

1.11 Compléments

Nous vous donnons encore quelques informations complémentaires utiles pour clore ce chapitre.

Il est judicieux d'effacer l'écran avant d'y afficher les premiers résulats. Ceci est possible avec cette instruction :

Instruction 9 :

nn CLS

Il est souvent utile de stoper l'exécution d'un programme lorsque, par exemple, on a programmé ce que l'on appelle une boucle infinie. En frappant la touche ESC on interrompt le traitement du programme. N'importe quelle autre touche, permet de reprendre ce traitement.

En actionnant deux fois la touche ESC, le programme s'arrête définitivement.

Instruction 10 :

nn STOP

Cette instruction permet à l'intérieur même du programme d'en arrêter le déroulement. La reprise du traitement du programme interrompu se fait par l'emploi de :

Commande 7 :

CONT

Pour conclure, il faut noter qu'il est souvent très utile d'ajouter des commentaires à un programme (titres, explications, etc.). Cela se fait grâce à l'instruction :

Instruction 11 :

nn REM Text

Cette instruction n'influe pas sur le fonctionnement du programme, mais figure sur toutes les listes qui en sont dressées.

Ce chapitre devrait suffire à donner à un débutant les bases du langage de programmation BASIC. Il se trouve maintenant à même de comprendre les programmes d'application utilisés par la suite. S'il devait rester quelques problèmes de compréhension, un conseil :

T E S T E R

La meilleure façon d'apprendre est de tester les programmes et d'essayer de comprendre les raisons pour lesquelles l'ordinateur se comporte comme il le fait.

MATHEMATIQUES



2.1 Avant-Propos

Si il y un domaine où l'ordinateur excelle, c'est bien celui des mathématiques. C'est une des raison de l'importance de l'implantation de tant d'ordinateurs. A tout seigneur, tout honneur, notre premier chapitre d'études portera sur les mathématiques. Nous aurons l'occasion de voir plus tard qu'elles ne ne sont qu'un domaine d'application parmi beaucoup d'autres.

Quoiqu'il en soit, le problème est de savoir, pour les prochains chapitres, comment rendre utiles les exemples que nous allons prendre. Les mathématiques comprennent, on le sait, un éventail très large de degrés de difficulté allant des opérations élémentaires aux calculs différentiels et aux intégrales.

Nous traiterons ici des problèmes de difficulté moyenne. Leurs énoncés seront donc semblables à ceux que l'on rencontre par exemple au lycée. En effet, ils se prêtent très bien à la démonstration des possibilités qu'offrent les ordinateurs et ceci sans que leur développement ne soit trop encombrant de part leur complexité. L'objet de ce livre n'est pas de fatiguer le lecteur par des exemples d'utilisations d'ordinateurs, mais de lui montrer ses possibilités. C'est la raison pour laquelle les énoncés sont structurés de manière simple et adaptés à la pratique quotidienne des plus jeunes lecteurs.

Note à propos des programmes de ce livre :

Aucun contrôle n'est fait sur les éventuelles erreurs de saisie dûes à l'utilisateur.

Par exemple, lorsqu'un problème mathématique n'utilise que des nombres positifs il n'a pas été prévu, dans le programme correspondant, de signaler à l'utilisateur son erreur s'il a entré un nombre négatif.

Nous allons maintenant commencer par un exemple très simple.

2.2 Le théorème de Pythagore

Pour chaque exemple, nous exécuterons les étapes de travail suivantes :

Etapes de travail

1 - Présentation du problème
2 - Analyse du problème
3 - Organigramme
4 - Programme
5 - Liste des variables
6 - Description du programme
7 - Résultats

Le théorème de Pythagore permet d'illuster clairement comment s'enchaînent ces différentes étapes.

1) Présentation du problème

Le théorème de Pythagore s'énonce ainsi :

Dans un triangle rectangle, le carré de l'hypothénuse est égale à la somme des carré des deux autres cotés, ou, en d'autres termes :

$$C^2 = A^2 + B^2$$

ou

$$C = \sqrt{A^2 + B^2}$$

2) Analyse du problème

L'analyse de ce problème est très simple :

Comme dans tous les cas qui suivront, il est nécessaire de se souvenir que, d'une manière tout à fait générale, une procédure de traitement de données se déroule selon le Principe **ETS** déjà présenté.

E = Entrée
T = Traitement
S = Sortie

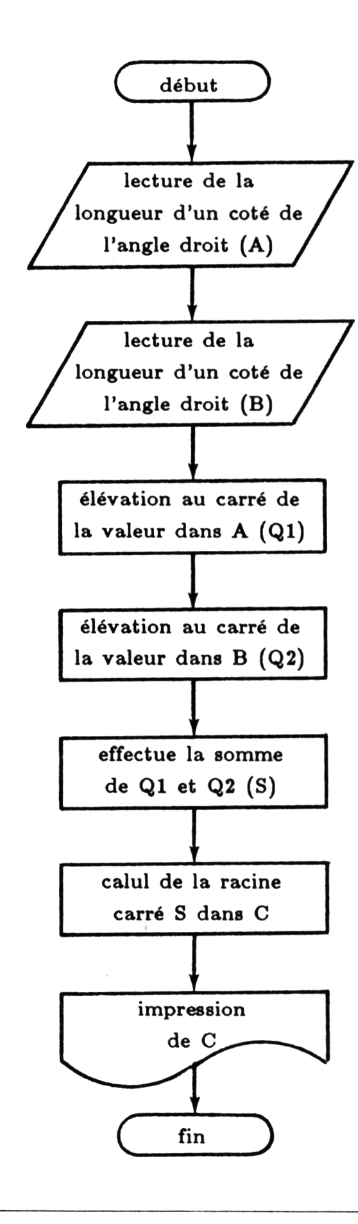
Pour l'analyse de ce problème il nous faut d'abord déterminer les informations que nous devons donner à l'ordinateur pour qu'il puisse résoudre le problème posé.

Les informations nécessaires sont naturellement les longueurs des deux côtés de l'angle droit A et B.

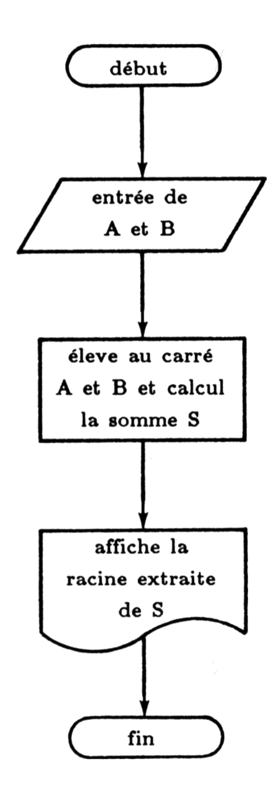
Au cours de l'étape du traitement, ces deux valeurs doivent être élevées au carré, puis additionnées. Il reste ensuite à en extraire la racine carrée.

L'étape de sortie se réduit ici à l'affichage à l'écran du résultat du calcul (la longueur de l'hypothénuse C).

3) L'organigramme



Version plus courte de l'organigramme



4) Le programme

```
10 REM M1-PYTHAGORE
20 CLS
30 PRINT "PROGRAMME DE CALCUL DE L'HYPOTHENUSE D'UN TRIANGLE RECTANGLE
"
40 PRINT
50 PRINT:PRINT:PRINT TAB (10)"PROF.W.VOSS. 1984":PRINT:PRINT
60 INPUT "PREMIER COTE DE L'ANGLE DROIT :";A
70 INPUT "DEUXIEME COTE DE L'ANGLE DROIT :";B
80 Q1=A*A
90 Q2=B*B
100 S=Q1+Q2
110 C=SQR(S)
120 PRINT:PRINT:PRINT"HYPOTHENUSE  C =";C
130 END
```

Version plus courte du programme

```
10 a 50 Comme ci-dessus
60 INPUT "COTES DE L'ANGLE DROIT :";A,B
70 C=SQR(A*A+B*B)
80 PRINT:PRINT"HYPOTHENUSE C =";C:END
```

5) Liste des variables

A	: Premier côté de l'angle droit
B	: Deuxième côté de l'angle droit
C	: Hypothénuse
Q1	: Carré du premier côté de l'angle droit A
Q2	: Carré du deuxième côté de l'angle droit B
S	: Somme des carrés des côtés de l'angle droit

6) Description du programme

Pour cette description, nous nous référons à la première version du programme :

Ligne 10 : Commentaire

Ligne 20 : L'écran est effacé

Ligne 30 - 50 : Impression du titre à l'écran

Ligne 60 : Une valeur est demandée pour le premier côté de l'angle droit du triangle et est stockée dans la zone A.

Ligne 70 : le second côté correspond à la zone B

Ligne 80 : calcul dans la zone A du carré de la valeur contenue dans la zone A.

Ligne 90 : calcul du second carré de la valeur contenue dans la zone B.

Ligne 100 : Calcul de la somme des deux carrés des côtés.

Ligne 110 : Détermination de la racine à partir de la somme des carrés.

Ligne 120 : Impression des résultats sur l'écran

Ligne 130 : Fin du programme

7) Résultats

Après le lancement du programme (première version du programme) apparaît ce titre sur l'écran :

PROGRAMME DE CALCUL DE L'HYPOTHENUSE
D'UN TRIANGLE RECTANGLE

Deux lignes plus bas apparait la première demande :

PREMIER COTE DE L'ANGLE DROIT : ?

Entrons, par exemple, la valeur 3. Il apparait alors à l'écran :

DEUXIEME COTE DE L'ANGLE DROIT : ?

Entrons par exemple la valeur 4, l'ordinateur répond alors :

HYPOTHENUSE C = 5

Ce programme permet donc de calculer l'hypothénuse de n'importe quel triangle rectangle.

Nous avons traité cet exemple en détail, pour illustrer la manière idéale pour résoudre un problème.
Nous abrégerons nos explications dans les exemples qui suivent, tout au moins pour ceux qui sont aussi simples que "le théorème de Pythagore".

2.3 PGCD et PPCM

1) Présentation du problème

Pour de nombreux calculs algébriques, et en particulier pour ceux qui contiennent des fractions, il est très utile de connaître le plus grand commun diviseur (PGCD) et le plus petit commun multiple (PPCM) de différents nombres.

Le PGCD est la plus grande valeur par laquelle deux nombres peuvent être divisés sans qu'il y ait de reste.

Quant au PPCM, il s'agit du plus petit nombre pouvant être divisé par deux valeurs données sans qu'il y ait de reste.

Si nous prenons, par exemple, les deux nombres Z1 = 30 et Z2 = 40, le PGCD est bien évidemment 10 et le PPCM prend la valeur 120 (le lecteur peut le vérifier par rapide calcul mental).

2) Analyse du problème

Comment, à partir de deux nombres donnés, trouve-t-on le PGCD et le PPCM ?

Essayons, à partir des nombres 30 et 40 de trouver le PGCD de la meilleure façon qui soit :

1ère étape : 40 / 30 = 1 reste 10

2ème étape : 30 / 10 = 3 reste 0

Dans cet exemple nous trouvons le PGCD de la manière suivante :

1ère étape : Division du plus grand nombre par le plus petit, notation du reste.

2ème étape : Division du plus petit nombre par le reste, notation du nouveau reste.

Si le nouveau reste est égale à zéro, le diviseur qui a conduit à ce reste nul, correspond au PGCD recherché.

Donnons un second exemple significatif :

Zone 1 : Z1 (le plus grand nombre)	Zone 2 : Z2 (le plus petit nombre)	Division	Reste
78	42	1	36
42	36	1	6
36	6	6	0

Le PGCD de 78 et de 42 est donc 6. Le schéma de calcul présenté dans le schéma ci-dessus, peut être facilement transformé en un organigramme.

Consacrons-nous maintenant à la recherche du plus petit commun multiple (PPCM) :

On obtient le PPCM des deux nombres Z1 et Z2 en divisant le produit de ces deux nombres par le PGCD.

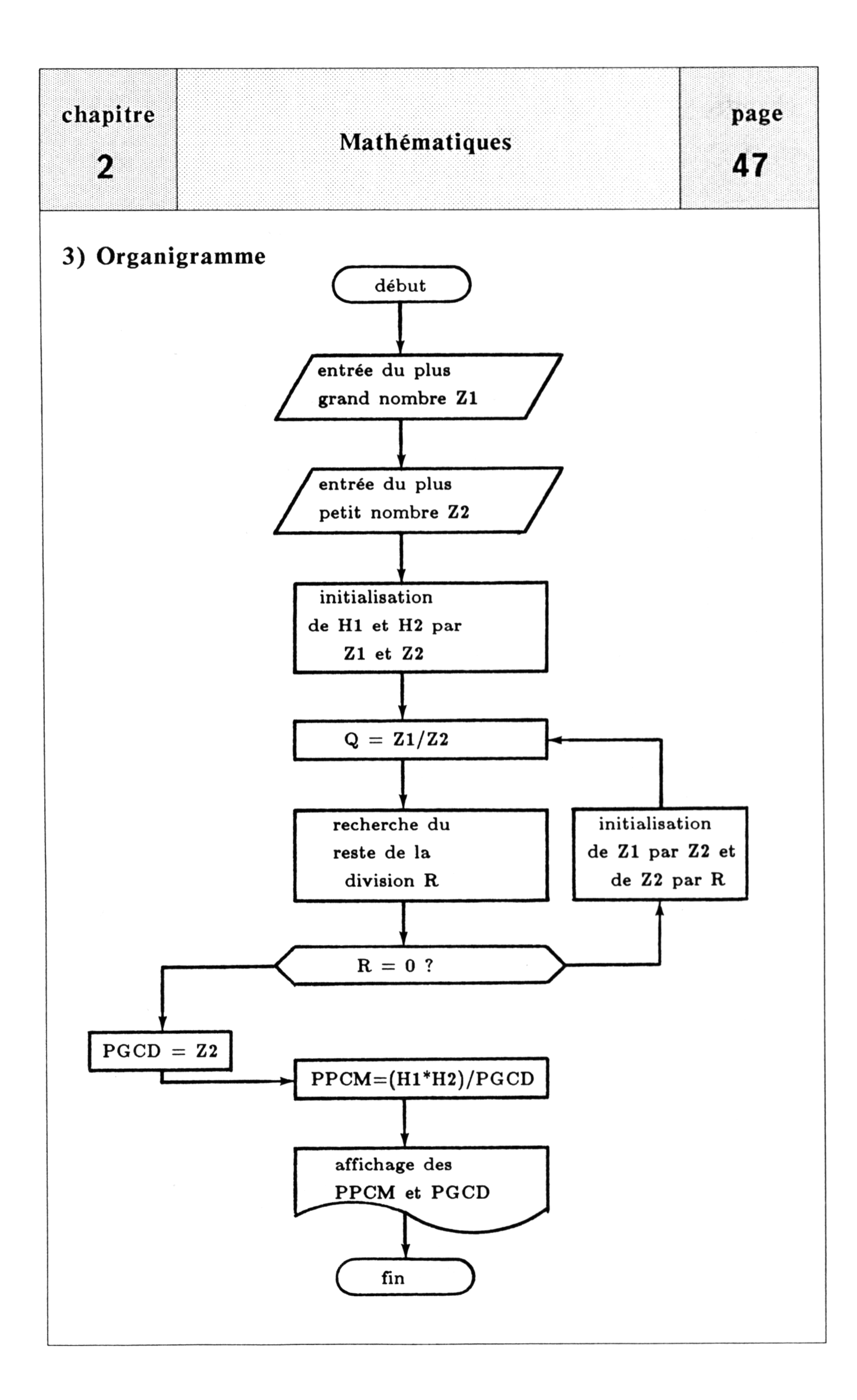
D'où :

Z1 = 78 et Z2 = 42

PPCM = (Z1 * Z2) / PGCD = (78 * 42) / 6 = 546

Dans un programme BASIC si nous souhaitons exécuter ce calcul, d'après le tableau général donné plus haut, il faut changer constamment les contenus des zones. Les valeurs à entrer Z1 et Z2 doivent être temporairement stockées en deux zones auxilaires H1 et H2 avant de procéder au calcul du PGCD pour être disponible lors de celui du PPCM.

3) Organigramme



4) Programme

```
10 REM M2-PGCD ET PPCM
20 CLS
30 PRINT "PROGRAMME POUR LA RECHERCHE DU PGCD ET PPCM"
40 PRINT "A PARTIR DE DEUX NOMBRES DONNES"
50 PRINT:PRINT:PRINT TAB (10)"PROF.VOSS. 1984":PRINT
60 INPUT "PLUS GRAND NOMBRE    : ";Z1
70 PRINT:PRINT
80 INPUT "PLUS PETIT NOMBRE    : ";Z2
90 H1=Z1:H2=Z2
100 Q=Z1/Z2
110 R=Z1-Z2*INT(Q)
120 IF R>0 THEN Z1=Z2:Z2=R:GOTO 100
130 PGCD=Z2
140 PPCM=(H1*H2)/PGCD
150 PRINT:PRINT:PRINT"PGCD =";PGCD
160 PRINT:PRINT:PRINT"PPCM =";PPCM
170 PRINT:PRINT"FIN DE CALCUL":END
```

5) Liste des variables

PGCD	: plus grand commun diviseur
H1	: zone auxiliaire 1
H2	: zone auxiliaire 2
PPCM	: plus petit commun multiple
Q	: quotient des deux nombres Z1 et Z2
R	: reste de la division
Z1	: plus grand nombre
Z2	: plus petit nombre

6) Description du programme

Ligne 10-50 : titre et explications
Ligne 60 : entrée d'un nombre
Ligne 70 : saut de deux lignes
Ligne 80 : entrée d'un deuxième nombre (inférieur au premier)
Ligne 90 : Initialisation des deux zones auxilaires H1 et H2 par les valeurs entrées, afin qu'elles soient disponibles par la suite (se reporter à la ligne 140).
Ligne 100 : Calcul du quotient des nombres entrés
Ligne 110 : Calcul du reste de la division au moyen de la fonction INT.
Ligne 120 : Si ce reste est supérieur à zéro, la zone Z1 est écrasée par la zone Z2 contenant le second nombre et la zone Z2 par la zone R contenant le reste. Le programme revient à la ligne 100.
Ligne 130 : Si au contraire, le reste est égal à zéro, le PGCD contenu dans Z2 est copié dans la zone PGCD.
Ligne 140 : Conformément à nos estimations le PPCM se détermine ensuite à partir du produit des deux nombres initiaux (qui se trouvent toujours dans les zones H1 et H2), divisé par le PGCD.
Ligne 150-160 : Impression des résultats séparés par des sauts de ligne.
Ligne 170 : Fin du programme.

7) résultats du programme

Après l'exécution de la commande RUN, le programme efface l'écran et affiche :

PROGRAMME POUR LA RECHERCHE DU PGCD
ET DU PPCM A PARTIR DE DEUX NOMBRES DONNES

PLUS GRAND NOMBRE : ?

Si, après le point d'interrogation, qui indique une interruption du programme, nous entrons par exemple la valeur 78, le programme réclame le second nombre :

PLUS PETIT NOMBRE : ?

Si nous entrons alors le nombre 42 il s'affiche alors à l'écran :

PGCD = 6
PPCM = 546
FIN DE CALCUL

2.4 Contrôle des nombres premiers

1) Présentation du problème

Il s'agit d'élaborer un programme contrôlant si un nombre entier positif quelconque est un nombre premier ou non.

On sait déjà que l'on entend par nombre premier un nombre qui n'est divisible que par 1 et par lui-même, de telle sorte que le reste soit nul.

2) analyse du problème

Pour vérifier si un nombre quelconque est premier il faut le diviser par tous les nombres compris entre 1 et lui-même. Si il y a un reste ou si le résultat de la division n'est pas un nombre entier, il faut reprendre l'opération avec un autre diviseur.

Si au cours de ces divisions on parvient à un résultat qui n'a pas de reste, on en déduit que le nombre dont il est question n'est pas premier.

Par contre, si on obtient un reste pour toutes les divisions, il faut en conclure que le nombre initial est un nombre premier.

La liste des diviseurs qu'il faut tester consciencieusement les uns après les autres commence par la valeur 2 et se termine par la valeur égale à la moitié du nombre initial, il n'est pas nécessaire de tester les diviseurs élevés car il n'est pas possible, avec l'un d'eux, de trouver un résultat entier.

Exemple : nombre initial X = 25

1er diviseur : 2 — 25 / 2 = 12 reste 1
2e diviseur : 3 — 25 / 3 = 8 reste 1
3e diviseur : 4 — 25 / 4 = 6 reste 1
4e diviseur : 5 — 25 / 5 = 5 reste 0

25 n'est pas un nombre premier.

Exemple : nombre initial X = 7

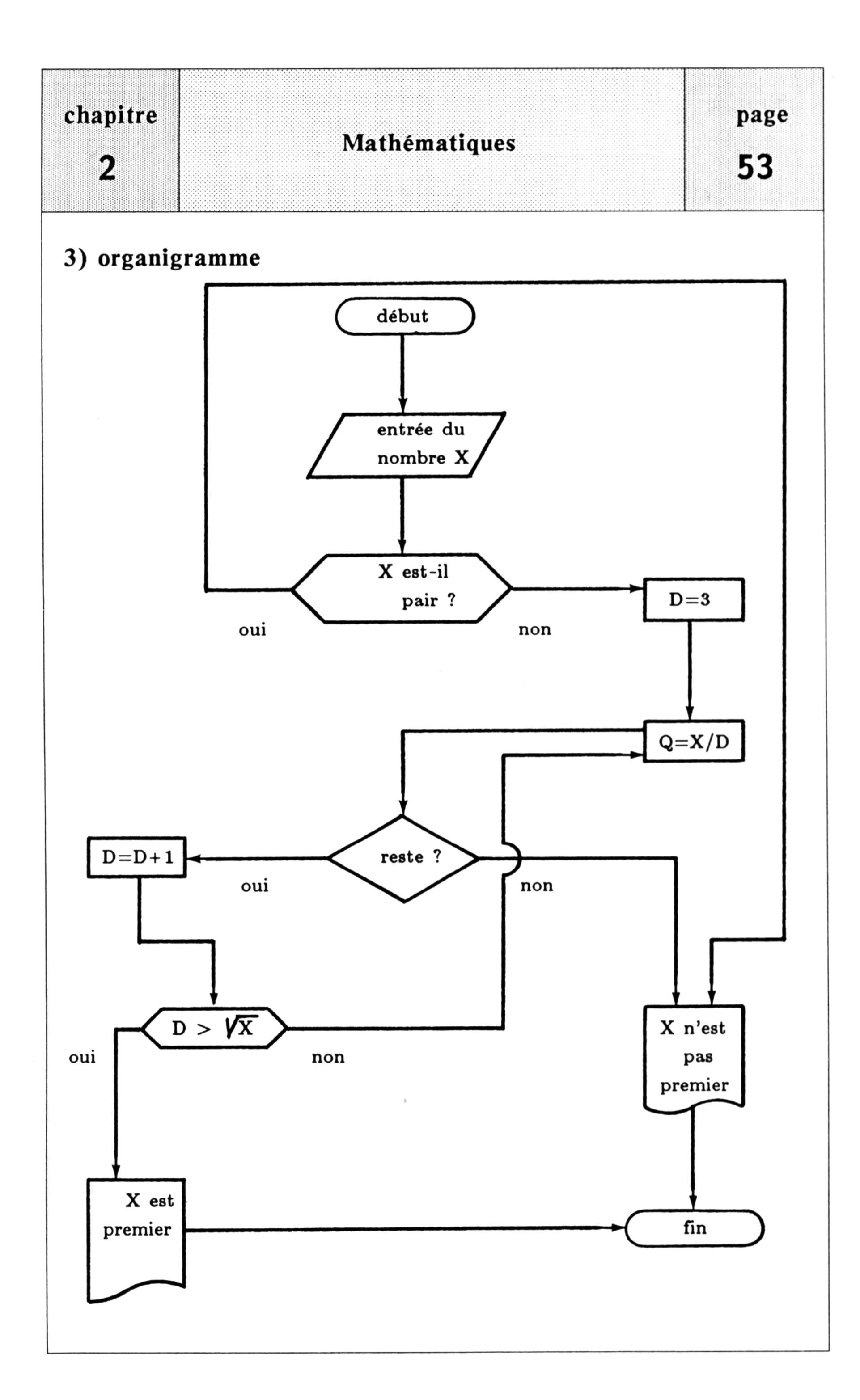
1er diviseur : 2 — 7 / 2 = 3 reste 1
2e diviseur : 3 — 7 / 3 = 2 reste 1

Il n'est pas utile de continuer nos tests puisqu'il n'est plus possible que le résultat de la division soit un nombre entier.

Puisque, dans cet exemple, on n'a pas trouvé de reste nul, on peut dire que :

7 est un nombre premier.

Ces deux exemples montrent que nous avons dû augmenter successivement le diviseur d'une unité, en commençant par la valeur 2, jusqu'à ce que l'on ait atteint la valeur maximale de X / 2.


3) organigramme
début
entrée du nombre X
X est-il pair ?
oui
non
D=3
Q=X/D
reste ?
D=D+1
oui
non
D > √X
oui
non
X n'est pas premier
X est premier
fin

Cet exemple montre clairement qu'en refléchissant soigneusement au problème, il est possible de simplifier de beaucoup le programme et donc d'accélérer son exécution :

On peut dès maintenant remarquer qu'un nombre pair ne peut pas être un premier puisqu'il est obligatoirement divisible par 2.

Ce qui fait que l'on peut écarter les nombres pairs comme étant des "nombres non-premiers" dès le début du programme.

Pour tout nombre impair, le diviseur maximum jusqu'auquel il est nécessaire d'aller n'est pas X/2 mais le nombre entier inférieur à $\sqrt{X}$. S'il n'a pas été possible d'obtenir, jusqu'à ce diviseur, une division sans reste, il ne le sera pas non plus avec des nombres plus élevés.

4) Programme

```
10 REM M3-NOMBRES PREMIERS
20 CLS
30 PRINT "PROGRAMME VERIFIANT SI UN NOMBRE ENTRE"
40 PRINT "EST PREMIER OU NON"
50 PRINT:PRINT TAB (10)"PROF.W.VOSS. 1984":PRINT
60 INPUT "ENTREZ UN NOMBRE ENTIER  :";X
70 T=2
80 IF X/2=INT (X/2) THEN 140
90 Q=X/T
100 IF Q=INT (Q) THEN 140
110 T=T+1
120 IF T<=SQR(X) THEN 90
130 PRINT:PRINT:PRINT X;" EST UN NOMBRE PREMIER.":GOTO 150
140 PRINT:PRINT:PRINT X;" N'EST PAS UN NOMBRE PREMIER"
145 PRINT "MAIS DIVISIBLE PAR";T
150 PRINT:PRINT:PRINT"FIN DES CALCULS.":END
```

5) Liste des variables

Q = Quotient du nombre X par le Diviseur D
D = Diviseur
X = Nombre à entrer qui sera traité pour savoir si c'est ou non un nombre premier.

6) Description du programme

Ligne 10-50 : Titre du programme, effacement de l'écran puis impression du titre.
Ligne 60 : Demande du nombre à tester.
Ligne 70 : Le diviseur D prend la valeur 2
Ligne 80 : Si le reste de la division de X par 2 est nul (alors X/2 = INT (X/2)), continuer à la ligne 90.
Ligne 90 : Division de X par D (ce qui donne la valeur de Q).
Ligne 100 : Si la division n'a pas de reste (auquel cas Q = INT (Q)), continuer à la ligne 140, sinon aller à la ligne 110.
Ligne 110 : Incrémentation de 1 de la valeur du diviseur D
Ligne 120 : Tant que la valeur D reste inférieur à $\sqrt{X}$ revenir à la ligne 90, sinon continuer à la ligne 130.
Ligne 130 : On arrive à cette ligne parce qu'aucune division n'a pu être effectuée sans reste. Ceci signifie que X est un nombre premier. Ce résultat affiché, il s'effectue un saut à la la ligne 150.
Ligne 140-150 : On n'arrive à cette ligne que si une division sans reste a été effectuée. Cela signifie alors que X n'est pas un nombre premier mais qu'il est divisible (au moins) par la valeur contenue dans la variable D.
Ligne 150 : Fin du programme.

7) Résultats

Dès lors, les titres des exemples ne seront plus repris.

Après l'affichage du titre à l'écran, le programme réclame le nombre qui doit être contrôlé et le stocke dans la zone X. Comme les titres, les phrases qui aident l'utilisateur à répondre à cette demande ne seront plus retranscrites.

Entrons par exemple, le nombre 13, l'ordinateur annonce alors :

```
13 EST UN NOMBRE PREMIER.
FIN DES CALCULS
```

Entrons maintenant le nombre 27, l'ordinateur annonce cette fois :

```
27 N'EST PAS UN NOMBRE PREMIER MAIS DIVISIBLE PAR 3.

FIN DES CALCULS.
```

2.5 Equation du second degré

1) Présentation du problème

La résolution d'équations du second degré fait partie des principaux problèmes posés en mathématiques en classe de 3ème. C'est une bonne introduction pour aborder des problèmes plus complexes.

La formule générale d'une équation du second degré, x étant l'inconnue à déterminer, est la suivante :

(1) $ax^2 + bx + c = 0$

En divisant cette équation par le coefficient a, on obtient :

(2) $x^2 + px + q = 0$

La résolution de l'équation mène à deux valeurs x1 et x2, comme le montre la formule suivante :

(3) $x1 = - p / 2 + \sqrt{p^2 / 4 - q}$

et

$x2 = - p / 2 - \sqrt{p^2 / 4 - q}$

2) Analyse du problème

Selon les différentes valeurs possibles des coefficients a, b et c la formulation des solutions mène à divers résultats.

Quand, par exemple, la valeur correspondant à l'expression qui se trouve sous la racine (discriminant D) est négative, nous obtenons des résultats qui n'ont rien à voir avec des valeurs réelles.

On peut aussi obtenir des solutions particulières lorsque le coefficient a = 0, mais dans ce cas on né se trouve plus vraiment en présence d'une équation du second degré.

C'est la raison pour laquelle nous n'aborderons que les cas "normaux" (a <> 0) et ceux qui sont résolvables (D > 0). Pour les valeurs particulières, le programme affichera un avis d'abandon.

Pour la résolution du problème, nous emploierons la formule (3) modifiée en formule (4) :

(4) $\mathbf{x1 = H + \sqrt{D}}$

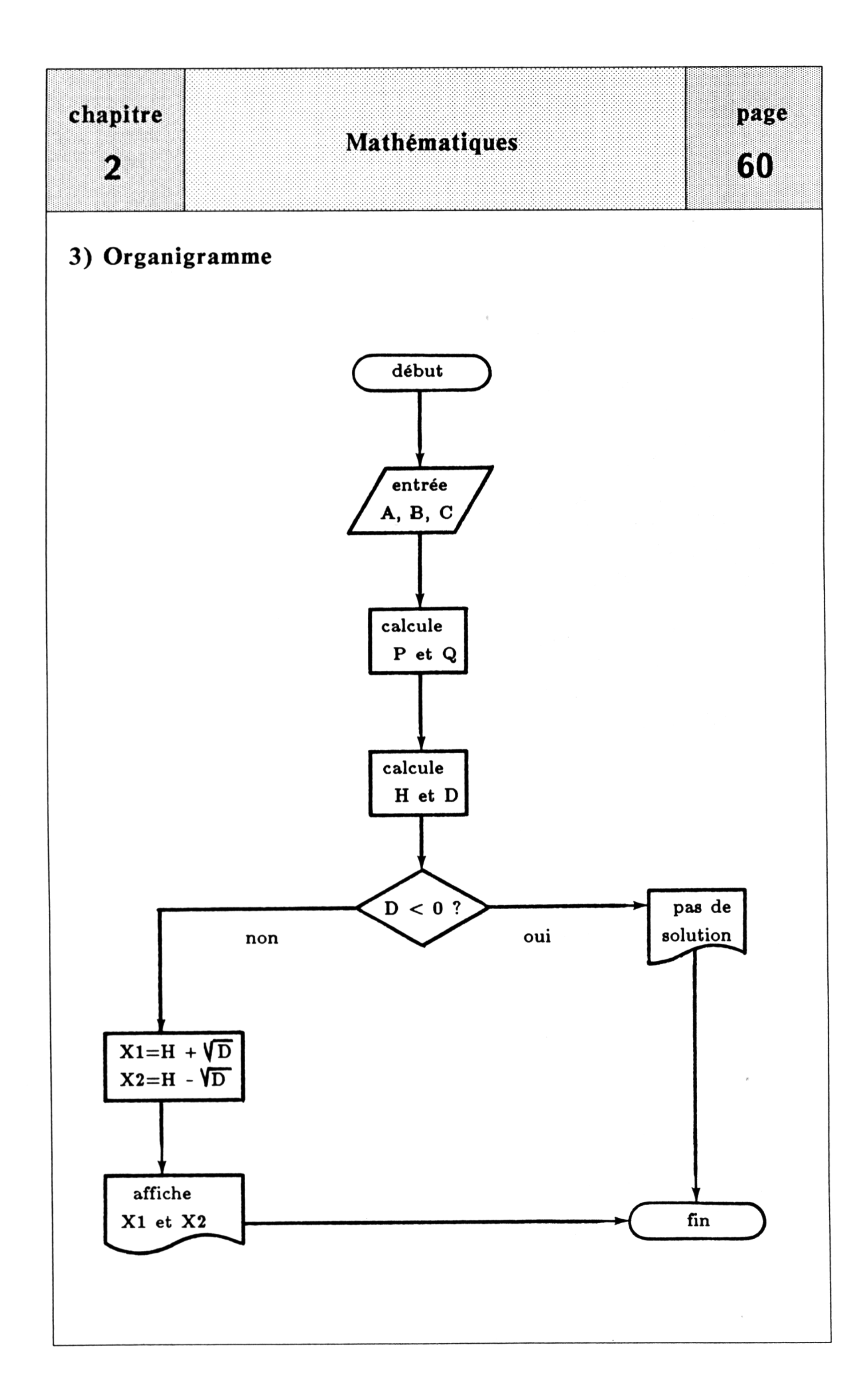
et

$\mathbf{x2 = H - \sqrt{D}}$

H est égal à - p / 2

Le programme BASIC correspondant doit donc d'abord déterminer les valeurs de D et de H afin de pouvoir donner les solutions ou un avis d'abandon selon que D > 0 ou D < 0.

3) Organigramme



4) Programme

```
10 REM M4-EQUATION DU SECOND DEGRE
20 CLS
30 PRINT"PROGRAMME DE RESOLUTION D'EQUATION"
40 PRINT"DU SECOND DEGRE"
50 PRINT:PRINT:PRINT TAB (10)"PROF.W.VOSS 1984":PRINT
60 INPUT "TERME DU SECOND DEGRE     A:";A
70 INPUT "TERME LINEAIRE            B:";B
80 INPUT "CONSTANTE DE L'EQUATION   C:";C
90 P=B/A:Q=C/A
100 H=-P/2:D=P*P/4-Q
110 IF D<0 THEN PRINT:PRINT "PAS DE SOLUTION ":GOTO 150
120 X1=H+SQR(D):X2=H-SQR(D)
130 PRINT:PRINT:PRINT"PREMIERE SOLUTION   X1=";X1
140 PRINT:PRINT:PRINT"DEUXIEME SOLUTION   X2=";X2
150 PRINT:PRINT:PRINT"FIN DES CALCULS.":END
```

5) Liste des variables

A = coefficient du terme du second degré
B = coefficient du terme linéaire
C = constante de l'équation du second degré

D = Discriminant
H = - P / 2
P = B / A
Q = C / A

X1 = première solution
X2 = deuxième solution

6) Description du programme

Ligne 10-50 : Commentaire et affichage du titre
Ligne 60-80 : Entrée des coefficients de l'équation (tel que dans l'équation (1)).
Ligne 90 : Détermination de P et de Q d'après l'équation (2).
Ligne 100 : Détermination de la valeur H et du discriminant D.
Ligne 110 : Si D est inférieur à zéro, cela entraîne l'affichage à l'écran d'un avis d'abandon et un saut à la ligne 150.
Ligne 120 : Si, par contre, le discriminant n'est pas négatif, les deux solutions X1 et X2 seront calculées.
Ligne 130-140 : Les deux solutions sont affichées à l'écran.
Ligne 15 : Fin du programme.

7) Résulats

Entrons par exemple :

A = 4
B = 4
C = - 24

On se trouve donc en présence de l'équation du second degré suivante :

$$4\,x^2 + 4\,x - 24 = 0$$

Les résultats donnés par le programme sont :

```
X1 = 2
X2 = -3
FIN DES CALCULS.
```

2.6 Nombre d'Euler

1) Présentation du problème

Le nombre d'Euler, noté **e**, encore appelé constante de croissance, joue un rôle très important dans les développements en séries mathématiques que l'on peut considérer, en ce qui concerne le contenu, comme étant des chaînes de croissance ou de reproduction.

Il est encore à la base de ce que l'on appelle les logarithmes népériens et nous le retrouverons dans le chapitre concernant les problèmes économiques.

En cas d'oubli de la valeur de ce nombre, il est possible de la calculer à l'aide d'un programme BASIC adéquat, pourvu que l'on sache comment on obtient ce nombre.

e est la valeur limite des grandeurs suivantes,

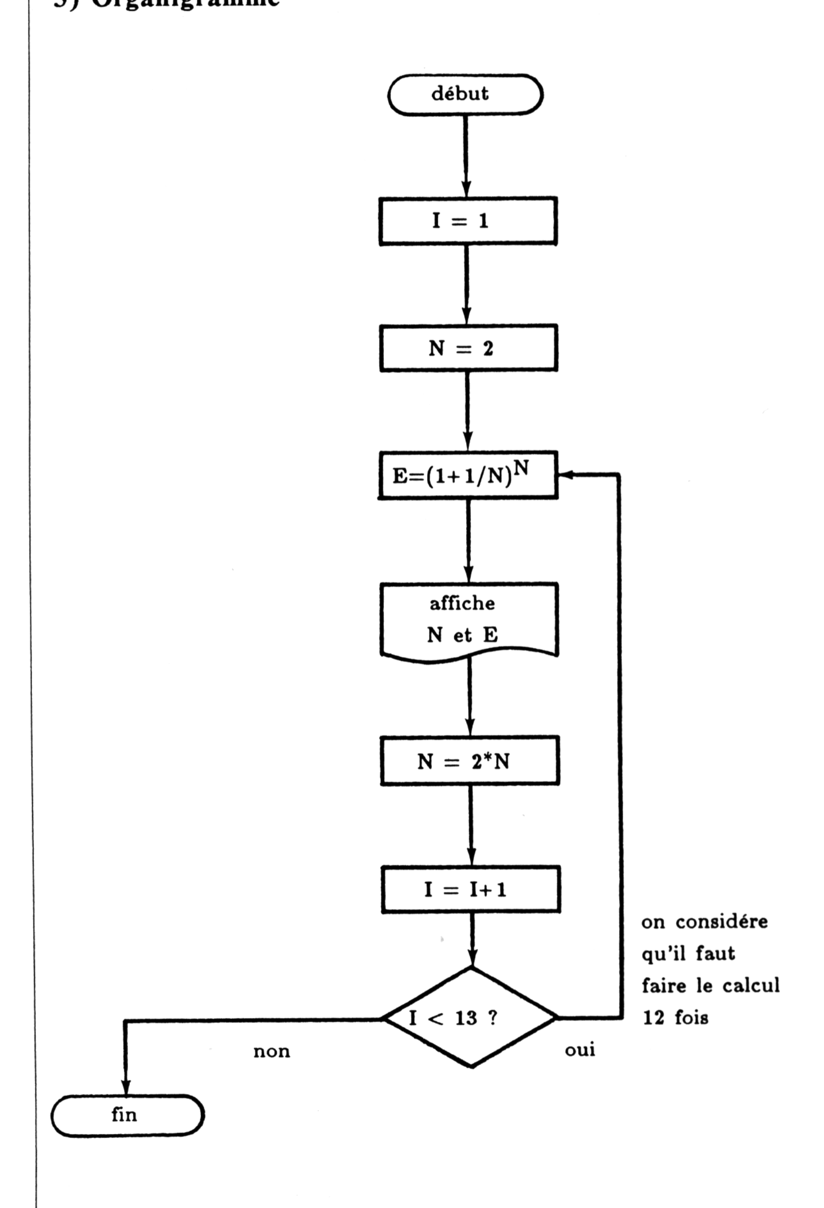
$e = (1 + 1/n)^n$ pour un nombre n tendant vers l'infini

2) Analyse du problème

Dans cet énoncé, l'analyse du problème est très simple, il suffit en effet de programmer la formule qui précède.

Il est possible de lancer ce programme plusieurs fois pour des valeurs de n, à chaque fois plus élevées. Comme la valeur de e n'est atteinte que pour n tendant vers l'infini, il est conseillé d'augmenter n rapidement, en doublant, par exemple, sa valeur à chaque opération.)

3) Organigramme



4) Programme

```
10 REM M5-NOMBRE D'EULER
20 CLS
30 PRINT"PROGRAMME DE CALCUL DU NOMBRE D'EULER"
40 PRINT
50 PRINT:PRINT:PRINT TAB(10)"PROF.W.VOSS. 1984":PRINT
60 N=2
70 PRINT"   N","    E":PRINT
80 FOR I=1 TO 12
90 E=(1+1/N)^N
100 PRINT N,E
110 N=N*2
120 NEXT I
130 PRINT:PRINT"FIN":END
```

5) Liste des variables

E = Nombre d'Euler (valeur approchée)
I = Indice de boucle (compteur)
N = Paramètre de la formule de calcul donné pour le nombre d'Euler

6) Description du programme

Ligne 10-50 : titre
Ligne 60 : Initialisation du paramètre N à la valeur 2
Ligne 70 : Impression de N et E. Saut de ligne
Ligne 80-120 : Boucle de 12 itérations. Pour chacune d'entre elles, E est calculé, N et E sont affichés et N est multiplié par deux.
Ligne 130 : Fin du programme.

7) Résultats

Le programme édite un tableau, les valeurs de E obtenues par calcul sont apposées aux valeurs croissantes de N.

En comparant ces valeurs, on s'aperçoit que **e** se rapproche d'un nombre fixe.

Pour avoir une valeur limite encore plus précise, le programme devrait utiliser une boucle plus large. Il faudrait alors modifier la ligne 80 comme suit :

```
80 FOR I = 1 TO 30
```

Mais il convient alors de prendre en considération le fait que les imprécisions dans les calculs que fait l'ordinateur peuvent mener à des erreurs si les valeurs de n sont trop élevées.

Le tableau résultant du programme cité plus haut est le suivant :

N	E
2	2.25
4	2.44140625
8	2.56578451
16	2.63792850
32	2.67699013
64	2.69734495
128	2.70773901
256	2.71299162
512	2.71563199
1024	2.71695572
2048	2.71761848
4096	2.71795007

FIN

2.7 Calcul des intérêts

1) Présentation du problème

La règle de trois joue un rôle prépondérant dans le calcul des intérêts. Elle permet par exemple de répondre à la question : à combien s'élève un montant M pris dans une somme totale T connaissant la valeur d'un montant M' pris dans une somme totale T'. Un exemple permettra de rendre plus clair cet énoncé.

Une facture s'élève à un montant de 450,00 F net. Le client doit payer en plus la taxe à la valeur ajoutée qui s'élève à 18,6 % du montant net de la facture. Il en résulte :

montant brut = montant net + montant de la TVA

Le problème d'arithmétique qui se pose ici, consiste donc à déterminer le montant de la TVA puis le montant brut, le montant net et le pourcentage de TVA étant donnés.

On pourrait aussi, bien sûr, demander de trouver le montant net en donnant le taux de TVA et le montant brut.

2) Analyse du problème

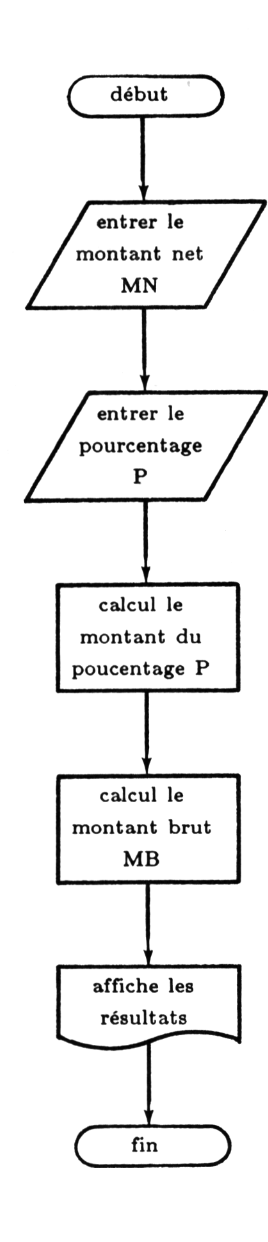
L'exemple donné peut être résolu grâce à la règle de trois :

montant net	= 100 % correspondant à 450 F
montant de la TVA	= 18,6 % correspondant à x F

x = (450 * 18.6) / 100 = 87.70 F

Pour ce faire nous devons entrer dans le programme le montant net et la TVA (ou un pourcentage quelconque).

3) Organigramme



4) Programme

```
10 REM M6-POURCENTAGE
20 CLS
30 PRINT"PROGRAMME DE CALCUL DES POURCENTAGES"
40 PRINT
50 PRINT:PRINT:PRINT TAB (10)"PROF.W.VOSS. 1984":PRINT
60 INPUT "MONTANT NET    :";NB
70 PRINT:INPUT "POURCENTAGE    :";SS
80 SB=NB*SS/100
90 BB=NB+SB
95 PRINT:PRINT
100 PRINT:PRINT"MONTANT NET            :";NB
110 PRINT:PRINT"MONTANT DU POURCENTAGE :";SB
120 PRINT:PRINT"MONTANT BRUT           :";BB
130 PRINT:PRINT"FIN":END
```

5) Liste des variables

BB = Montant brut

NB = Montant net

SB = Montant du pourcentage

SS = Pourcentage

6) Description du programme

Ligne 10-50 : titre
Ligne 60-70 : Entrée du montant net et du poucentage
Ligne 80-90 : Calcul du montant du pourcentage et du montant brut
Ligne 95 : saut de deux lignes
Ligne 100-120 : sortie des résultats sur l'écran
Ligne 130 : Fin du programme

7) Résultats

Si par exemple, on entre la valeur 200 pour le montant net et la valeur 15 pour le pourcentage, on aura à l'exécution :

```
MONTANT NET                    : 200
MONTANT DU POURCENTAGE         : 30
MONTANT BRUT                   : 230
FIN
```

2.8 Dés

1) Présentation du problème

Nous allons présenter un programme qui lance 100 fois les dés et qui calcule la moyenne des nombres sortis.

Pour simuler le lancement des dés on peut se servir de la fonction RND (RND = Random = hasard) qui génère des nombres aléatoires compris entre 0 et 1 inclus).

On appelle nombres aléatoires les nombres qui apparaissent dans une suite irrégulière et non ordonnée.

2) Analyse du problème

Dans le cas de l'AMSTRAD CPC il est possible d'utiliser la fonction RND sans argument.

Il est recommandé au lecteur de la tester en entrant par exemple plusieurs fois :

PRINT RND

valider après chaque entrée en tapant sur la touche ENTER

FOR I = 1 TO 10 : PRINT RND : NEXT puis ENTER

Apparaissent alors à l'écran des nombres aléatoires compris entre 0 et 1.

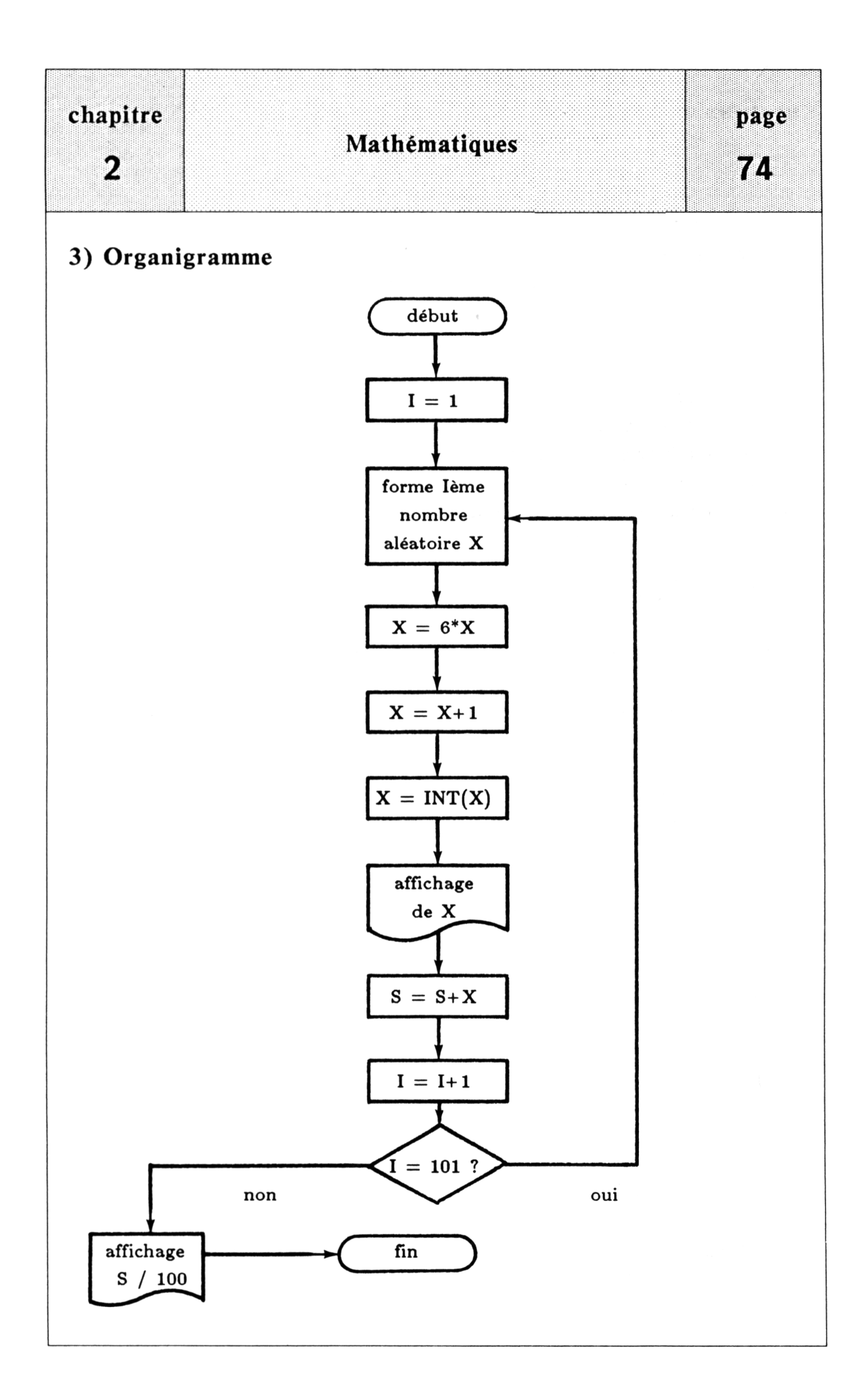
Cherchons maintenant à obtenir 100 valeurs entières comprises entre 1 et 6 correspondant aux différentes valeurs des dés.

Le schéma suivant indique les étapes à suivre afin de les obtenir.

Traitement	**Résultat**
Fonction RND	entre 0 et 1
*6	entre 0 et 6 non compris
+1	entre 1 et 7 non compris
Fonction INT	entiers entre 1 et 6 compris

Il nous reste donc à calculer la moyenne de ces nombres en en faisant la somme puis en la divisant par 100.

3) Organigramme



4) Programme

```
10 REM M7-DES
20 CLS
30 PRINT"PROGRAMME SIMULANT 100 FOIS LE LANCEMENT"
40 PRINT"D'UN DES ET CALCUL DE LA MOYENNE"
50 PRINT"DES RESULTATS"
60 PRINT:PRINT:PRINT
70 PRINT TAB (10)"PROF.W.VOSS 1984":PRINT
80 I=1
90 X=RND
100 X=X*6:X=X+1:X=INT(X)
110 PRINT X;
120 S=S+X
130 I=I+1
140 IF I<101 THEN 90
150 AM = S/100
160 PRINT:PRINT:PRINT
170 PRINT:PRINT"MOYENNE :";AM
180 PRINT:PRINT"FIN DES CALCULS":END
```

5) Liste des variables

AM = Valeur moyenne

I = Indice de passage (compteur)

S = Somme de la totalité des résultats donnés par les dés

X = Résultat aléatoire

6) Description du programme

Ligne 10-70 : titre
Ligne 80 : initialisation du compteur
Ligne 90-100 : génération d'un nombre aléatoire et transformation de ce nombre en une valeur entière comprise entre 1 et 6.
Ligne 110 : Affichage à l'écran de ce nombre X
Ligne 120 : Addition de ce nombre X à la somme S des résultats de dés déjà sortis
Ligne 130 : Incrémentation de 1 du compteur I
Ligne 140 : Tant que I est inférieur à 101, saut à la ligne 90.
Ligne 150-170 : Calcul de la valeur moyenne et sortie à l'écran de cette valeur
Ligne 180 : Fin du programme

7) Résultats

Après avoir lancé le programme on voit apparaître à l'écran son titre, 100 lignes comprenant chacune un résultat aléatoire correspondant au lancement de dé (on laissera au lecteur le soin de réfléchir au fait que ces résultats figurent sur des lignes différentes) puis la moyenne calculée à partir des 100 résultats.

Celle-ci, si le générateur de nombres aléatoires fonctionne normalement, doit se situer aux alentours de 3.5.

Ces résultats étant aléatoires il n'est pas utile à titre d'exemple d'en donner une liste.

2.9 Problèmes de la précision dans les calculs

Un ordinateur tel que l'AMSTRAD CPC ne réserve pour chacun des nombres réels qu'un nombre donné d'octets. Cela signifie que la précision avec laquelle il travaille est limitée.

On s'en rend très bien compte en visualisant de "grandes" valeurs initialisées au sein d'un programme.

Exemple :

```
10  REM PRECISION DU CALCUL
20  A = 12345678
20  B = 123456789
40  C = 1234567891
50  D = 12345678912
60  ?A:?B:?C:?D
70  END
```

Résultats :

```
12345678
123456789
1.234567 E + 09
1.234567 E + 10
```

Ainsi, si l'ordinateur reçoit des nombres comportant plus de 9 chiffres (exemple C et D), il ne peut les redonner que sous une forme normalisée avec exposant. Ce qui donne, par exemple, pour **e** :

1.234567E+09

Cette écriture correspond au nombre .

$1.234567 * 10^9 = 1.234567 * 1\ 000\ 000\ 000$

Soit :

1 234 567 000

La valeur de C est donc tronquée après son septième chiffre, et sa valeur initiale est définitivement perdue.

Le véritable problème est donc de prendre en considération ce phénomène de tronquage pour effectuer correctement des calculs sur des valeurs sujettes à de telles coupures.

Illustrons ce problème :

Dans l'exemple précédent incrémentons de 1 la valeur de D. Nous avons donc une ligne supplémentaire au programme et nommons E cette nouvelle valeur :

```
55 E = D + 1
```

Faisons imprimer la valeur e à la ligne 65 :

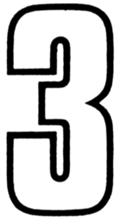
Soit :

```
65 ?E
```

Nous n'obtenons pas de différence entre les valeurs de D et de E l'ordinateur fait donc une erreur de calcul !

Conclusion : Il faut être très attentif lorsque l'on doit manipuler de grandes valeurs.

CHIMIE



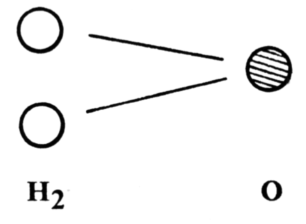
3.1 Instructions supplémentaires BASIC

Pour établir des programmes de chimie on a tout intérêt à exploiter les possibilités graphique et couleur qu'offre le CPC, pour la représentation de molécules ou de combinaisons chimiques.

On peut représenter la molécule de l'eau par :

H_2O

qui, on le sait, se compose de deux atomes d'hydrogène (H) et d'un d'oxygène (0) :



H_2 **O**

Pour arriver à de telles représentations, il est nécessaire d'employer certaines instructions. Celles-ci concernent d'une part, la génération et le changement de couleurs au cours de l'exécution d'un programme et d'autre part, l'utilisation de symboles spécifiques (Ce ne sera que dans un chapitre ultérieur que nous serons amenés à parler des possibilités qu'offre le graphisme haute résolution).

L'ensemble des caractères graphiques de l'AMSTRAD CPC se prête remarquablement bien à la fabrication de tels dessins (se reporter au manuel d'utilisation).

Ces symboles, qui ne figurent pas nécessairement sur le clavier ne peuvent être appelés que par leur code ASCII 126 comme d'ailleurs quelques caractères spéciaux et certaines lettres de l'alphabet grec). Pour cela on utilise la fonction suivante, déjà mentionnée :

CHR$ (argument)

L'argument de cette fonction est le numéro de code ASCII du caractère que l'on veut obtenir.

Si l'utilisateur entre par exemple :

PRINT CHR$ (231)

un symbole ressemblant à une petite boule apparait sur l'écran.

Voici à cette occasion quelques indications au sujet de la génération des couleurs pour ceux qui utilisent une télévision couleur. Il faut noter que les programmes présentés dans cet ouvrage utilisent des effets de couleur qui fonctionnent aussi avec les appareils noir et blanc. Les différentes couleurs prennent alors diverses nuances de vert ou sont hachurés.

A sa mise en marche l' AMSTRAD CPC se trouve en mode 1 (40 caractères par ligne) et dans le cas du moniteur couleur, l'affichage se fait en jaune sur fond bleu.

Les couleurs peuvent être modifiées par l'instruction suivante :

Instruction 12 :

nn PEN nombre

Grâce à cette instruction, il est possible de représenter quatre couleurs différentes en se conformant aux indications suivantes :

Nombre	**Couleur**
1	jaune
2	bleu clair
3	rouge
4	bleu

La dernière couleur n'est bien sûr pas exploitable sur un fond bleu !

Le programme suivant permet de visualiser trois petites boules de couleurs différentes les unes au dessous des autres :

```
10 FOR I = 1 TO 3
20 PEN I
30 PRINT I, CHR$(231)
40 NEXT I
50 END
```

Remarque :

Si au cours de l'expérimentation des couleurs l'utilisateur se trouve bloqué, il a la possibilité de réinitialiser l'ordinateur en pressant simultanément les touches :

CRTL SHIFT et ESC

ATTENTION : le programme qui se trouvait dans l'ordinateur est perdu.

Nous emploierons désormais quelques instructions qui permettront à l'ordinateur de recevoir un plus grand nombre d'informations et ceci d'une manière plus adroite qu'avec les instructions LET ou INPUT.

L'instruction **DATA** fait partie de ces instructions si elle est associée à l'instruction **READ**.

Instruction 13 :

nn DATA valeur 1, valeur 2, valeur 3....

Instruction 14 :

nn READ variable 1, variable 2, variable 3...

Exemple :

```
10 REM ENTREE DES DONNEES
20 DATA 75000, PARIS, 13000, MARSEILLE
30 DATA 69000, LYON, 38000, GRENOBLE
40 READ P1,01$,P2,02$,P3,03$,P4,04$
50 ?P1,01$:?P2,02$:?P3,03$:?P4,04$
60 END
```

Nous demandons au lecteur d'essayer ce programme.

Pour les fichiers de données de grande taille, il est recommandé de n'utiliser qu'un seul nom de variable qui comprenne alors plusieurs valeurs. Cela est possible en utilisant des variables indicées, telles qu'on les connait en mathématiques par exemple on écrit :

X_1, X_2, X_3, ... pour tout X_i

Il en est de même en BASIC :

X(1), X(2), X(3),...pour tout X(I)

(syntaxe : nom de variable (nom de l'indice))

Mais pour employer ces variables indicées il est nécessaire d'indiquer à l'ordinateur, au début du programme correspondant, combien d'emplacements mémoire doivent leur être réservées.

C'est l'instruction **DIM** qui s'en charge :

Instruction 15 :

nn DIM nom de var.1(Nombre1),nom de var.2(nombre2),...

Par exemple cette instruction

10 DIM X(20)

réserve 21 places mémoire pour la variable X. On parle de tableau à une dimension.

Il s'agit là des places X(0), X(1), X(2), X(3),...,X(20)

Il est aussi possible d'utiliser des tableaux à deux dimensions (bidimentionnels)

L'instruction :

10 DIM Z(2,3)

réserve à la variable Z (tableau Z) 3*4 = 12 places que l'on peut répartir sur trois lignes et quatre colonnes. On peut les représenter ainsi :

Z(0,0)	Z(0,1)	Z(0,2)	Z(0,3)
Z(1,0)	Z(1,1)	Z(1,2)	Z(1,3)
Z(2,0)	Z(2,1)	Z(2,2)	Z(2,3)

L'écriture générale d'un élément d'un tel tableau est :

Z (I,J)

Le premier indice (ici I) est l'indice de ligne, le second (ici J) est l'indice de colonne.

Pour conclure, il faut préciser que l'on accède toujours à une instruction DATA à l'aide d'une l'instruction READ. Cela signifie qu'une seconde instruction READ continue à lire le DATA actuel à l'endroit où le READ précèdent s'est arrêté.

Néanmoins, il est souvent utile de relire la liste DATA depuis le début. Cela se fait au moyen de l'instruction suivante :

Instruction 16 :

nn RESTORE

Cette instruction a pour effet de renvoyer au début de la première instruction DATA l'index de lecture de telle manière que les données peuvent être relues en utilisant à nouveau l'instruction READ.

3.2 La molécule d'eau

1) Présentation du problème

Il s'agit d'obtenir une représentation graphique de la molécule d'eau (H_2 O) à l'aide d'un programme BASIC (se référer aussi au paragraphe 3.1.). Les atomes d'hydrogène devront être représentés par de petites boules jaunes, les atomes d'oxygène par des boules rouges.

Ce programme servira de base pour les problèmes plus complexes que posent les réprésentations graphiques.

2) Analyse du problème

Si l'on se représente le graphique que doit créer le programme l'analyse du problème est très simple.

Ce graphique est identique à celui déjà dessiné au début de ce chapitre.

Nous voyons que ce graphique se compose de quatre lignes :

1ère ligne : Une petite boule jaune et un trait oblique de haut, à gauche, en bas à droite.

2ème ligne : Une petite boule rouge (déplacée à droite).

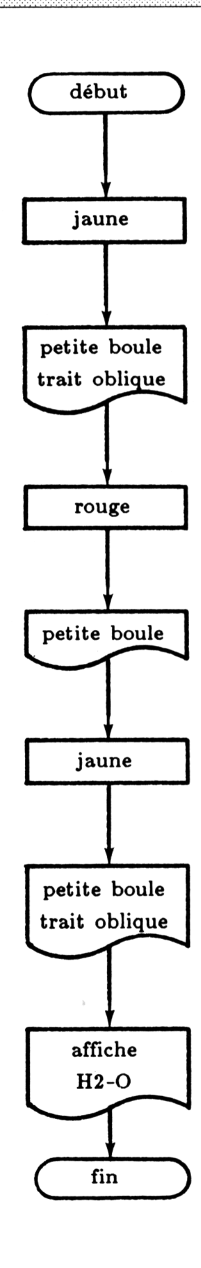
3ème ligne : Une petite boule jaune sous la première ligne et un trait oblique du bas à gauche vers le haut à droite.

4ème ligne : Texte : H_2 O.

Nous obtenons la petite boule comme précédemment au moyen de la fonction CHR$ avec le caractère du code ASCII portant le numéro 231.

En conséquence, le trait oblique partant du haut à gauche vers le bas à droite (\) a le numéro de code 205 et le trait oblique opposé (/) le numéro de code 204.

3) Organigramme



4) Programme

```
10 REM C1-H2O
20 CLS
30 PRINT"PROGRAMME POUR LA REPRESENTATION GRAPHIQUE"
40 PRINT"DE LA MOLECULE D'EAU  :  H2O"
50 PRINT:PRINT:PRINT TAB(10)"PROF.W.VOSS 1984":PRINT
60 PRINT:PRINT:PRINT
70 PEN 1:PRINT TAB(10) CHR$(231);
80 PRINT TAB(12) CHR$(205)
90 PEN 3:PRINT TAB(14) CHR$(231)
100 PEN 1:PRINT TAB(10) CHR$(231);
110 PRINT TAB(12) CHR$(204)
130 PRINT:PRINT TAB(10)"H2 -O"
140 PRINT:PRINT:PRINT"FIN":END
```

5) Liste des variables

Aucune variable n'est utilisée dans ce programme.

6) Description du programme

Ligne 10-60 : titre
Ligne 70 : Passage au jaune et affichage d'une petite boule dans la colonne 10, on reste à la même ligne
Ligne 80 : Sortie d'un trait oblique dans la colonne 12
Ligne 90 : Passage au rouge et sortie d'une petite boule dans la colonne 14
Ligne 100 : Passage au jaune et sortie d'une petite boule dans la colonne 10, on reste à la même ligne
Ligne 110 : Sortie d'un trait oblique dans la colonne 12
Ligne 130 : Sortie de H_2O à partir de la colonne 10 après un saut de deux lignes.
Ligne 140 : Fin du programme

7) Résultats

La sortie de ce programme sur l'écran correspond à l'image que nous avons présentée au début du chapitre 3.

3.3 Equation d'une réaction chimique

Les réactions chimiques peuvent, elles aussi, être représentées graphiquement. On peut le voir en prenant l'exemple des procédures obligatoires relatives à la production de l'acier.

Dans les pays industrialisés, la production de l'acier commence par celle de la fonte brute dans les hauts fourneaux. Elle se déroule en deux temps :

1èr temps : on fait se consumer l'air chauffé avec du carbone pour obtenir du monoxyde de carbone :

2 C +	**O_2**	**------------>**	**2 CO**
carbone	oxygène		monoxyde de carbone

2ème temps : Le monoxyde s'élève dans le haut fourneau et réagit au contact de l'oxyde de fer (Fe_2 O_3) :

Fe_2 O_3	**+**	**3 CO ------->**	**2 Fe +**	**3 CO_2**
oxyde de fer		monoxyde de carbone	fer	gaz carbonique

La fonte brute est ainsi prête pour d'autres utilisations.

Il s'agit maintenant de présenter un programme BASIC qui illustre les liaisons inhérentes à cette réaction chimique.

Pour représenter les atomes et les molécules, nous nous servirons des possibilités graphiques développées dans l'exemple précédent.

2) Analyse du problème

La description précédente et ce qui a été dit au sujet de l'exemple précédent nous dispensent de cette analyse. De plus longs développements seraient inutiles ici.

3) Organigramme

L'organigramme n'est pas non plus nécessaire (cf. 2ème étape).

4) Programme

```
10 REM C2-REACTION CHIMIQUE
20 CLS
30 PRINT"PROGRAMME DE REPRESENTATION"
40 PRINT"D'UNE REACTION CHIMIQUE"
50 PRINT:PRINT:PRINT TAB(10)"PROF.W.VOSS 1984":PRINT
60 PRINT:PRINT:PRINT
70 PEN 1:PRINT TAB(10) CHR$(231);
75 PRINT TAB(25) CHR$(231);:PEN 2:PRINT CHR$(231)
80 PRINT:PRINT TAB(12)"+  ";:PEN 2:PRINT CHR$(231);:PRINT CHR$(231);
85 PRINT"   =   "
86 PRINT
90 PEN 1:PRINT TAB (10) CHR$(231);
100 PRINT TAB (25) CHR$(231);:PEN 2:PRINT CHR$(231)
110 PRINT:PRINT
120 PEN 1:PRINT TAB (10)"2C + O    =    2CO"
130 PRINT TAB (16) "2"
140 PRINT:PRINT:PRINT:PRINT"ENTREZ LA COMMANDE <CONT> S.V.P":STOP
150 CLS
160 PEN 2:PRINT TAB (33)CHR$(231)
170 PEN 1:PRINT TAB (18)CHR$(231);:PEN 2:PRINT CHR$(231);
180 PRINT TAB(32)CHR$(231)
190 PEN 2:PRINT TAB(12) CHR$(231);:PEN 3:PRINT TAB (26) CHR$(231);:PEN
 2:PRINT TAB (33) CHR$(231)
200 PEN 3:PRINT TAB(10) CHR$(231);:PEN 2:PRINT TAB (33) CHR$(231)
210 PRINT TAB (12)CHR$(231);:PRINT TAB(15) "+";
220 PEN 1:PRINT TAB (18) CHR$(231);:PRINT TAB (19) CHR$(231);
230 PEN 1:PRINT TAB (23) "=";:PRINT TAB (29) "+";:PEN 1:PRINT TAB (32)
 CHR$(231)
240 PEN 3:PRINT TAB (10)CHR$(231);:PEN 2:PRINT TAB(33) CHR$(231)
250 PRINT TAB(12) CHR$(231);:PEN 3:PRINT TAB(26)CHR$(231);:PEN 2:PRINT
 TAB(33) CHR$(231)
260 PEN 1:PRINT TAB (18)CHR$(231);:PEN 2:PRINT TAB (19) CHR$(231);
270 PEN 1:PRINT TAB(32) CHR$(231)
280 PEN 2:PRINT TAB(33) CHR$(231)
290 PRINT:PRINT
```

```
300 PEN 1:PRINT TAB (8) "FE O   +  3CO  =  2FE + CO"
310 PRINT TAB (10)"2 3                     2"
320 PRINT:PRINT"FIN"
```

5) Liste des variables

Aucune variable n'a été utilisée.

6) Description du programme

Ligne 10-60	: titre
Ligne 70	: sortie d'une petite boule jaune dans la colonne 10
Ligne 75	: sortie d'une petite boule jaune dans la colonne 25 et d'une bleue à côté
Ligne 80	: sortie d'un caractère + dans la colonne 12 et de deux petites boules bleues à côté
Ligne 85	: sortie d'un caractère = à la même ligne
Ligne 86	: ligne vide
Ligne 90	: idem ligne 70
Ligne 100	: idem ligne 75
Ligne 110-130	: sortie de l'équation de la réaction
Ligne 140-150	: attente et effacement de l'écran
Ligne 160-280	: sortie de la représentation graphique de la seconde équation chimique
Ligne 290-310	: sortie de la seconde équation de la réaction
Ligne 320	: fin du programme

7) Résultats

Le programme présenté fait apparaître deux graphiques l'un après l'autre sur l'écran, le second se réalise lorsque l'utilisateur du programme a entré la commande CONT.

Il est inutile ici de représenter ces sorties graphiques puisque l'on a pas l'usage de la couleur.

Il est donc recommandé de tester ce programme directement sur un poste couleur.

3.4 Calcul stoechiométrique

1) Présentation du problème

Le calcul stoechiométrique consiste à établir les proportions suivant lesquelles les corps se combinent entre eux dans les réactions chimiques.

Combien de grammes de soufre faut-il pour transformer totalement 100 grammes de zinc en sulfure de zinc ? (soufre = S ; zinc = Zn ; sulfure de zinc = ZnS)

Poids atomique du soufre : 32.1
Poids atomique du zinc : 65.4

2) Analyse du problème

Pour résoudre le problème posé il est nécessaire de partir de la relation de réaction adéquate.

Zn + S -------> ZnS

Lorsque nous appliquons l'une à l'autre les masses moléculaires du corps recherché et du corps donné à l'avance, nous obtenons :

32.1 g de soufre sont nécessaires pour transformer 65.4 g de zinc (voir plus haut).

L'on peut dire maintenant :

x / 100 = 32.1 / 65.4

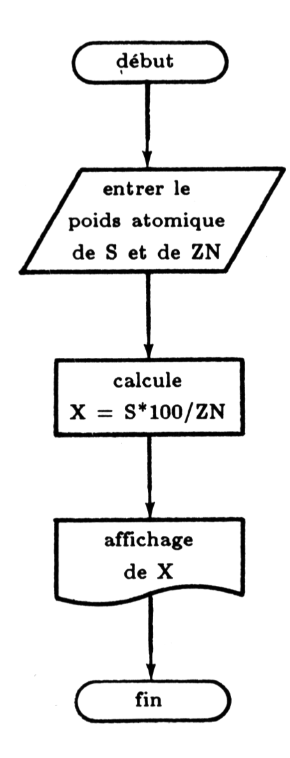
ou encore :

x = 3210 / 65.4

x est ici la valeur recherchée.

3) Organigramme

Une fois encore l'ordinogramme du problème est très simple :



4) Programme

```
10 REM C3-CALCUL DE STOECHIOMETRIE
20 CLS
30 PRINT"PROGRAMME REALISANT UN CALCUL STOECHIOMETRIQUE"
40 PRINT:PRINT
50 PRINT:PRINT:PRINT TAB (10) "PROF.W.VOSS 1984":PRINT
60 PRINT:PRINT:PRINT"EXEMLE :":PRINT:PRINT
70 PRINT"ZINC + SOUFRE = SULFURE DE ZINC :"
80 PRINT:PRINT"ZN   +   S    =  ZN S":PRINT:PRINT
90 PRINT:PRINT"COMBIEN FAUT IL DE GRAMME DE SOUFRE,"
100 PRINT"POUR TRANSFORMER TOTALEMENT 100g DE"
110 PRINT"ZINC      ?":PRINT:PRINT
120 INPUT "1er  ELEMENT : ";A$
130 INPUT "2eme ELEMENT : ";B$
140 PRINT:PRINT:PRINT
150 PRINT"POIDS ATOMIQUE DE ";A$;:PRINT TAB(30)":";:INPUT A
160 PRINT"POIDS ATOMIQUE DE ";B$;:PRINT TAB(30)":";:INPUT B
170 X=B*100/A
180 PRINT:PRINT:PRINT"QUANTITE UTILISEE DE ";B$;" : ":PRINT:PRINT TAB(
5) X;" G"
190 PRINT:PRINT:PRINT"FIN DU CALCUL":END
```

5) Liste des variables

A = Poids atomique du corps A
A$= Désignation du corps A

B = Poids atomique du corps B
B$ = Désignation du corps B

X = Proportion recherchée

6) Description du programme

Ligne 10-110 : titre et énoncée du problème à traiter
Ligne 120-130 : entrée des noms des deux corps
Ligne 140 : 3 lignes vides
Ligne 150-160 : sortie des noms des corps et demande de leur poids atomique
Ligne 170 : calcul de la quantité recherchée
Ligne 180 : affichage du résultat (plus quelques lignes vides sur lesquelles désormais nous n'attirons plus obligatoirement l'attention)
Ligne 190 : fin du programme

7) Résultats

Après avoir donné quelques indications, le programme demande le nom du premier élément :

Entrons par exemple "zinc", l'ordinateur réclame alors le nom du second élément qui entre dans la réaction :

Tapons "soufre", l'ordinateur attend les poids atomiques du zinc et du soufre.

Après quoi nous entrons les valeurs 32.1 et 65.4 et le programme affiche le résultat suivant :

QUANTITE DE SOUFRE UTILISEE :

203.738318 G

FIN DU CALCUL

Il va de soi que, le cas échéant, le programme doit être radicalement modifié pour d'autres processus chimiques (lorsque, par exemple, plus de deux corps interviennent dans la réaction en question).

3.5 Classification périodique des éléments

1) Présentation du problème

Nous savons désormais que l'ordinateur est capable de réaliser aussi bien des opérations que des représentations graphiques. Nous allons maintenant l'utiliser en tant que source de renseignements.

Nous prendrons l'exemple de la classification périodique des éléments.

L'utilisateur pourra apprécier, après simple entrée du nom d'un corps (ex : soufre), d'obtenir de l'ordinateur de nombreuses informations telles que :

1 - Symbole chimique
2 - Numéro atomique dans le classement périodique
3 - Poids atomique
4 - Poids spécifique
5 - Point de fusion
6 - Point d'ébullition
7 - Appartenance à un groupe (C-VIII,0)
8 - Appartenance à un sous-groupe (a,b)
9 - Couches électroniques (K-Q)

2) Analyse du problème

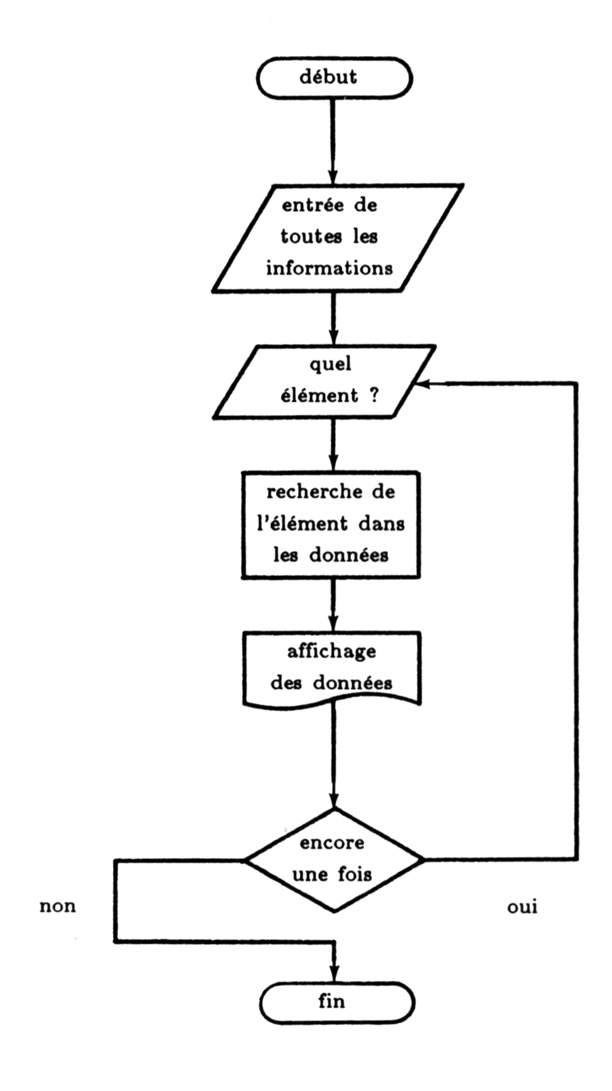
L'élaboration d'un programme qui fournisse une série d'informations sur certaines données ne pose ici aucune difficulté particulière.

Les informations générales peuvent être introduites par les fonctions DATA et READ. L'utilisateur interroge la base d'informations par l'instruction INPUT en donnant le nom du corps simple sur lequel il souhaite avoir des renseignements. Le programme met alors à sa disposition l'ensemble des données qu'il possède sur cet élément.

Enfin, il est possible de réinterroger la base d'informations pour un autre corps chimique.

3) Organigramme

L'ordinogramme présente la même simplicité que l'analyse du problème.



4) Programme

```
10 REM C4-ELEMENTS
20 CLS
30 PRINT"PROGRAMME DESTINE A INDIQUER LES PROPRIETES"
40 PRINT TAB(5) "ESSENTIELLES D'ELEMENTS CHIMIQUES"
50 PRINT:PRINT:PRINT TAB (10)"PROF.W.VOSS 1984":PRINT
60 N=52:REM NOMBRE D'ELEMENTS ENREGISTRES
70 DIM N$(N,10)
75 FOR J=1 TO 7
80 FOR I=1 TO N
85 READ N$(I,J)
90 NEXT I
95 NEXT J
100 FOR I=1 TO N
105 READ N$(I,8),N$(I,9),N$(I,10)
108 NEXT I
110 PRINT:INPUT "NOM DE L'ELEMENT S.V.P ";B$
120 FOR I=1 TO N:IF B$=N$(I,1) THEN 160
130 NEXT I
140 PRINT:PRINT"CET ELEMENT NE SE TROUVE PAS"
150 PRINT"DANS LE STOCK DES DONNEES":GOTO 290
160 CLS
170 PRINT N$(I,1);:PRINT TAB(23)N$(I,2)
180 PRINT:PRINT
190 PRINT"NUMERO ATOMIQUE          : ";N$(I,3)
200 PRINT"GROUPE                   : ";N$(I,8)
210 PRINT"SOUS-GROUPE              : ";N$(I,9)
220 PRINT"COUCHE                   : ";N$(I,10)
230 PRINT:PRINT
240 PRINT"POIDS ATOMIQUE           : ";N$(I,4)
250 PRINT"POIDS SPECIFIQUE         : ";N$(I,5)
260 PRINT
270 PRINT"POINT DE FUSION          : ";N$(I,6)
280 PRINT"POINT D'EBULLITION       : ";N$(I,7)
290 PRINT:PRINT:INPUT "UN AUTRE ELEMENT   (O/N) ";A$
300 IF A$="O" THEN CLS : GOTO 110
```

```
310 PRINT:PRINT"FIN EN LIGNE 310":END
495 REM DONNEES DES ELEMENTS
500 DATA ALUMINIUM,ANTIMOINE,ARGON,ARSENIC,BARYUM,PLOMB,BORE,BROME,CHL
ORE,CHROME
510 DATA FER,FLUOR,OR,HELIUM,IRIDIUM,IODE,CADMIUM,POTASSIUM,CALCIUM,CO
BALT
520 DATA CARBONE,CUIVRE,LITHIUM,MAGNESIUM,MANGANESE,MOLYBDENE,SODIUM
530 DATA NEON,NICKEL,PHOSPHORE,PLATINE,PLUTONIUM,MERCURE,RADIUM,RADON
540 DATA OXYGENE,SOUFRE,SELENIUM,ARGENT,SILICIUM,AZOTE
550 DATA STRONTIUM,THORIUM,TITANE,URANIUM,VANADIUM,HYDROGENE,BISMUTH,T
UNGSTENE
570 DATA XENON,ZINC,ETAIN
595 REM ABREVIATIONS DES ELEMENTS
600 DATA AL,SB,AR,AS,BA,PB,B,BR,CL,CR,FE,F,AU,HE,IR,I,CD,K,CA,CO,C,CU,
LI,MG
610 DATA MN,MO,NA,NE,NI,P,PT,PU,HG,RA,RN,O,S,SE,AG,SI,N,SR,TH,TI,U,V,H
620 DATA BI,W,XE,ZN,SN
640 REM NUMEROS ATOMIQUES DES ELEMENTS
650 DATA 13,51,18,33,56,82,5,35,17,24,26,9,79,2,77,53,48,19,20,27,6,29
660 DATA 3,12,25,42,11,10,28,15,78,94,80,88,86,8,16,34,47,14,7,38
670 DATA 90,22,92,23,1,83,74,54,30,50
695 REM POIDS ATOMIQUES DES ELEMENTS
700 DATA 26.97,121.76,39.94,74.91,137.36,207.21,10.82,79.92,35.46,52.0
1
710 DATA 55.85,19,197,4.003,192.2,126.92,112.41,39.096,40.08,58.94,12.
011
720 DATA 63.54,6.94,24.32,54.94,95.95,22.991,20.183,58.71,30.98,195.09
730 DATA 242,200.61,226.05,222,16,32.066,78.96,107.88,28.06,14.008
740 DATA 87.63,232.05,47.9,238.07,50.95,1.008,209,183.86,131.3,65.38,1
18.7
745 REM POIDS SPECIFIQUES DES ELEMENTS
750 DATA 2.7,6.67,.0018,5.72,3.5,11.34,1.73,3.14,1.557,7.14,7.86,.0017
760 DATA 19.25,.0018,22.42,4.942,8.64,.862,1.545,8.83,3.51,8.933,.534
770 DATA 1.75,7.3,10.2,0.971,0.0009,8.8,1.83,21.4,-,13.55,-,0.009,0.00
14,2.07
780 DATA 4.46,10.5,2.34,.00125,2.63,11.7,4.43,19,6.07,.0009,9.8,19.3
```

```
790 DATA .0059,7.14,7.28
795 REM POINTS DE FUSION DES ELEMENTS
800 DATA 658,630,-190,,704,327,2400,-7.3,-100,1800,1525,-218,1063,-272
810 DATA 2454,114,321,64,851,1490,-,1083,180,650,1221,2622,99
820 DATA -248,6,1452,44,1771,-,-39,700,-71,-218.7,113,220,960,1414
830 DATA -210,771,1827,1727,1689,1726,-262,271,3380,-111,9,419,232
845 REM POINT D'EBULITION DES ELEMENTS
850 DATA 2270,1440,-186,616,1537,1690,2550,58,-34,2600,2450,-187,2677
860 DATA -269,4400,184,770,762,1439,3185,-,2360,1372,1120,2152,4800,88
0
870 DATA -246,3075,280,3800,-,357,1140,-61.9,-182.97,444,688,1930,2630
880 DATA -195.8,1360,3530,-,-,3000,-272.78,1560,6000,-108.1,906,2430
900 REM CLASSE DES ELEMENTS
910 DATA III,B,M,V,B,O,0,A,M,V,B,N,II,A,P,IV,B,P,III,B,L,VII,B,N
920 DATA VII,B,M,VI,A,N,VIII,A,N,VII,B,L,I,B,P,0,A,K,VIII,A,P,VII,B,O
930 DATA II,B,O,I,A,N,II,A,N,VIII,A,N,IV,B,L,I,B,N,I,A,L,II,A,M,VII,A,
N
940 DATA VI,A,O,I,A,M,0,A,L,VIII,A,N,V,B,M,VIII,A,P,-,-,-,II,B,P,II,A,
Q,0,A,P
950 DATA VI,B,L,VI,B,M,VI,B,N,I,B,O,IV,B,M,V,B,L,II,A,O,-,-,-,IV,A,N
960 DATA -,-,-,V,A,N,I,A,K,V,B,P,VI,A,P,0,A,O,II,B,N,IV,B,O
```

5) Liste des variables

A$: Chaîne de caractères de réponse (Oui/Non)
B$: Variable contenant l'élément étudié par l'utilisateur
I : Indice de boucle (compteur)
J : Indice de boucle (compteur)
N : Nombre d'éléments de l'ensemble de données
N$: Elément du fichier

6) Description du programme

Ligne 10-50 : titre
Ligne 55 : message "un moment s'il vous plait" car la lecture de l'ensemble des données dure quelques secondes
Ligne 60 : Indication du nombre d'éléments. Dans ce programme il n'y a que 52 éléments chimiques enregistrés. Le cas échéant cette ligne doit être modifiée s'il y a changement du nombre des instructions DATA
Ligne 70 : dimensionnement du tableau contenant les informations
Ligne 75-108 : lecture de toutes les informations
Ligne 110 : demande de l'élément dont on souhaite avoir des renseignements
Ligne 120-130 : recherche de l'élément souhaité dans l'ensemble des données, s'il est trouvé, on continue à la ligne 160. S'il ne l'est pas poursuite à la ligne 140.
Ligne 140-150 : si l'élément souhaité ne figure pas dans l'ensemble des données, un message le signale puis il y a un saut à la ligne 290
Ligne 160 : effacement de l'écran
Ligne 170 : affichage à l'écran du nom de l'élément et du symbole chimique correspondant
Ligne 180 : saut de ligne
Ligne 190-280 : sortie à l'écran des autres informations concernant l'élément souhaité
Ligne 290-300 : interrogation pour savoir si une autre recherche d'informations sur un élément est désiré. Si oui effacement de l'écran et saut à la ligne 110. Si non, poursuite à la ligne 310
Ligne 310 : fin du programme
Ligne 495-960 : liste des données dans les instructions DATA

7) Résultats

Si par exemple, nous faisons une recherche sur le soufre, l'ordinateur répond :

SOUFRE	: S
NUMERO ATOMIQUE	: 16
GROUPE	: VI
SOUS GROUPE	: B
COUCHE	: M
POIDS ATOMIQUE	: 32.066
POIDS SPECFIQUE	: 2.07
POINT DE FUSION	: 113
POINT D'EBULLITION	: 444

UN AUTRE ELEMENT (O/N)?

PHYSIQUE



4.1 Avant-propos

Le domaine de la physique est comparable à celui des mathématiques. (se reporter au chapitre 2). Néanmoins quelques particularités peuvent s'ajouter à celles déjà rencontrées. Il est alors recommandé d'utiliser les possibilités graphiques de l'ordinateur.

C'est la raison pour laquelle nous allons compléter ce qui a déjà été dit, dans le chapitre précédent, sur les méthodes de programmation des couleurs et des graphiques. Ces instructions graphiques, en BASIC, sont développées dans la section suivante.

4.2 Programmation graphique

Un des aspects très intéréssant de l'AMSTRAD CPC est sans conteste le mode graphique haute résolution.

Selon le mode d'écran dans lequel travaille l'ordinateur, on dispose d'une capacité de résolution graphique différente (et comme on l'a déjà mentionné d'autres gammes de couleurs).

La commande suivante détermine le mode de l'écran :

Commande 8 :

MODE nombre

"nombre" peut prendre les valeurs zéro, un ou deux.

La signification de ces différentes valeurs est la suivante :

MODE	CARACTERES PAR LIGNE
0	20
1	40
2	80

Après sa mise en service, l'ordinateur se trouve, par défaut, en mode 1 et met donc à notre disposition 40 caractères par ligne d'écran.

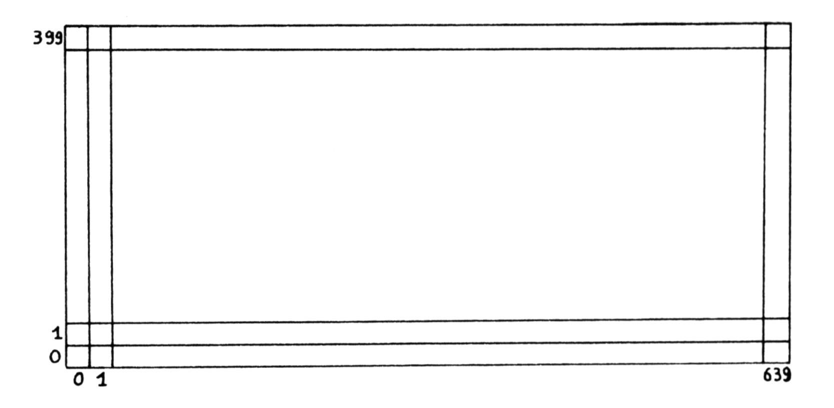
D'ordinaire, pour le graphisme de haute résolution, il possède une capacité de résolution s'élevant à 640 * 400 points.
(640 colonnes et 400 lignes)

Cette capacité de résolution permet des tracés relativement précis à l'aide des instructions adéquates.

Les plus importantes de ces instructions sont brièvement commentées par la suite et testées dans une série d'exemples simples et généraux.

Lorsque l'on produit des schémas en mode haute résolution il faut imaginer un curseur graphique (invisible) se déplacant sur l'écran, sa position indiquant l'endroit exact où l'on peut dessiner.

L'écran, en mode de graphique, peut donc être schématisé ainsi :



Il faut savoir que la ligne 0 se trouve tout en bas de l'écran et que la colonne 0 se trouve à son extrême gauche.

Ainsi le point (0,0), origine du système de coordonnées se situe en bas à gauche de l'écran comme on en a pris l'habitude dans les cours de mathématiques.

Si l'on veut afficher un point sur l'écran, on se sert de cette instruction :

Instruction 17 :

PLOT C,L

Les lettres C et L désignent respectivement la colonne (0-639) et la ligne (0-399) sur lesquelles on doit dessiner.

Il est conseillé au lecteur de tester le programme suivant :

```
10 CLS
20 PLOT 320,200
30 END
```

Le résultat est un petit point placé au centre de l'écran.

Tracer des lignes droites n'est pas plus compliqué, si l'on emploie l'instruction suivante :

Instruction 18 :

DRAW C,L

4.3 Programmes en BASIC

L'instruction DRAW 100,100, par exemple, tire un trait à partir de la position du curseur graphique, au moment où l'on fait cette opération, jusqu'au point de coordonnées (100,100).

Si le curseur graphique n'a été déplacé ni par une instruction PLOT ni pas une instruction DRAW, si donc il se trouve dans sa position d'origine, ce trait va de (0,0) à (100,100) ; (trait oblique d'environ 10 cm vers la droite et le haut de l'écran).

Admettons que nous voulons tracer une droite verticale passant par le centre de l'écran. Nous exécutons ce programme :

```
10 CLS
20 PLOT 320,0
30 DRAW 320,399
40 END
```

De la même manière, pour tracer une droite horizontale passant par le centre de l'écran nous utilisons le programme suivant :

```
10 CLS
20 PLOT 0,200
30 DRAW 639,200
40 END
```

Il est aussi facile de tracer un système de coordonnées, comme le montre cet exemple :

```
10 CLS
20 PLOT 0,0
30 DRAW 400,0
40 PLOT 0,0
50 DRAW 0,250
60 END
```

A l'inverse de bien des ordinateurs, l'insertion de textes en mode graphique se réalise facilement avec l'AMSTRAD CPC.

Il est possible, bien sûr d'élaborer des graphiques à partir de caractères spéciaux que le code ASCII met à notre disposition.

Ces caractères ont, dans le code ASCII, des numéros compris entre 123 et 255 et peuvent être appelés par l'instruction :

PRINT CHR$ (nombre) nombre = 123 à 255

Nous laissons au lecteur le soin de tester le programme suivant qui permet la visualisation à l'écran de tous ces caractères spéciaux. A peut près toutes les 15 lignes le programme s'interrompt pour que l'utilisateur puisse prendre le temps de regarder ces symboles. Il faut ensuite frapper sur le clavier la commande :

CONT pour remettre le programme en marche

```
10 CLS
20 PRINT"CARACTERES SPECIAUX":PRINT
30 PRINT
40 FOR I=123 TO 255
50 PRINT I;"     ";CHR$(I)
60 IF I/15 = INT (I/15) THEN STOP
70 NEXT I
80 END
```

L'utilisation de ces caractères spéciaux accessibles grâce à la fonction CHR$ est recommandée pour faire des dessins.

Par exemple, le numéro de code 124 nous permet de tracer un trait vertical de 5 mm environ.

Si maintenant nous programmons :

```
10 CLS
20 PRINT:PRINT:PRINT
30 FOR I=1 TO 10
40 PRINT TAB (10) CHR$(124)
50 NEXT I
60 END
```

Nous obtenons un trait vertical s'étendant sur 10 lignes, situé dans la colonne 10 de l'écran. En outre celui-ci se retrouve en mode graphique normal (40 colonnes et 25 lignes).

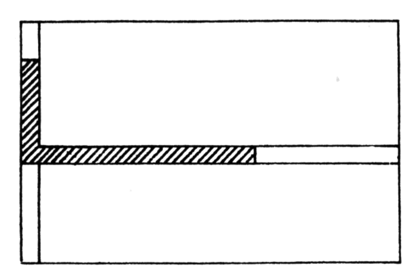
De la même manière, il est possible d'utiliser tous les autres symboles en employant la fonction CHR$.

Les dessins, à partir de ces symboles, sont également réalisables en mode 1, avec pour contrainte, leur visuallisation à l'écran sur 40 colonnes et 25 lignes.

Il n'est donc plus question de graphique haute résolution, comme le montre l'exemple suivant :

```
10 CLS
20 PRINT:PRINT:PRINT:PRINT
30 FOR I= 1 TO 10
40 PRINT CHR$(143)
50 NEXT I
60 FOR I= 1 TO 20
70 PRINT CHR$(143);
80 NEXT I
90 PRINT:PRINT:PRINT"FIN":END
```

Ce programme exécute le schéma suivant :



Il est préférable, dans ce genre de graphique de varier les couleurs en insérant les instructions :

15 PEN 2

et

85 PEN 1

Dans le programme ci-dessus le système de coordonnées en traits épais sortira en bleu ciel mais le mot "FIN" sera en jaune (couleur initiale).

Dans le programme précédent, le lecteur peut remarquer que la position basse des axes est dûe à la série d'instructions PRINT, placée en tête du programme.

Ce déplacement au positionnement en un point précis de l'écran, peut être obtenu simplement avec :

Instruction 19 :

nn LOCATE C,L

Cette instruction positionne le curseur au point de l'écran caractérisé par la colonne C (1-40) et la ligne L(1-25).

Utilisons cette instruction afin d'écrire le mot "ICI" au centre de l'écran :

```
10 CLS
20 LOCATE 19,12
30 PRINT "ICI"
40 FIN
```

Donc, grâce à cette instruction, nous allons maintenant pouvoir très simplement mêler graphiques et textes ce qui n'est pas toujours possible sur les autres ordinateurs, surtout en mode graphique haute résolution.

Si nous voulons, par exemple, nommer les axes du système de coordonnées par ordonnées (axe horizontal) et abscisses (axe vertical) exécutons ce programme :

```
10 CLS
20 LOCATE 1,10
30 PRINT"ORDONNEE"
40 LOCATE 25,22
50 PRINT"ABSCISSE"
60 PLOT 50,50
70 DRAW 350,50
80 PLOT 50,50
90 DRAW 50,220
100 END
```

Des représentations analogues sont envisageables à partir de ce modèle.

Après ces observations générales concernant la programmation graphiques revenons maintenant à des problèmes concrets.

4.4 Le peson à ressort

1) Présentation du problème

Lorsque l'on pend à un peson des objets de poids différents, celui-ci s'allonge plus ou moins.

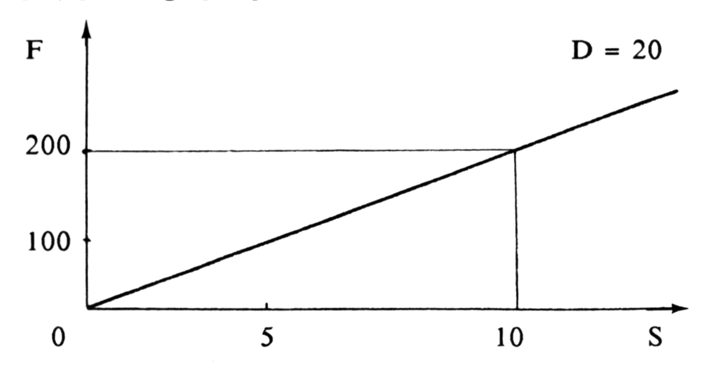
Il y a là une loi de proportionnalité qui veut que lorsque l'on double le poids, l'allongement du peson à ressort double également. Il existe donc une constante dans l'extension du ressort, à savoir le quotient de la force F exercée par le poids sur le peson et l'allongement du ressort S ; on peut donc dire que :

D = F/S = constante

Si D = 20 par exemple, cela veut dire qu'un poids de 100 (Kg) entraîne un allongement de 5 cm, un poids de 200 (Kg) un allongement de 10 cm, etc.

2) Analyse du problème

Le rapport décrit plus haut peut être représenté (pour D = 20 par exemple) par le graphique suivant :

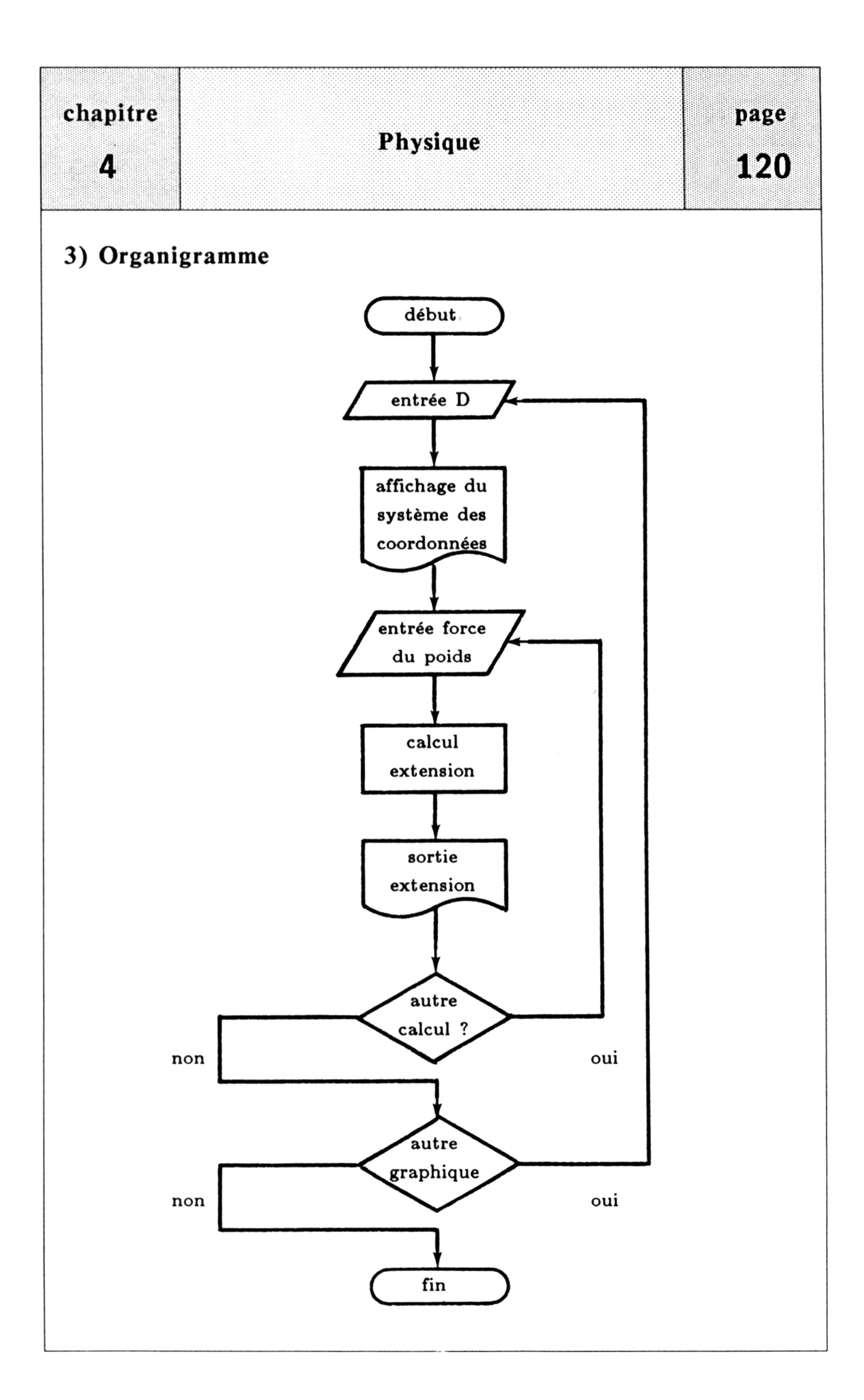


Le programme en BASIC que l'on se propose ici d'élaborer doit permettre de visualiser la représentation graphique correspondant à différentes valeurs de constante de ressorts.

Il comprend donc les étapes suivantes :

- formation d'un système de coordonnées,
- entrée d'une valeur pour la constante D
- tracé de la droite correspondant à cette constante (voir le schéma ci-dessus),
- entrée d'un poids quelconque et calcul de l'allongement du ressort en fonction de ce poids.

3) Organigramme



4) Programme

```
10 REM P1-PESON A RESSORT
20 CLS
30 PRINT"PROGRAMME EXPOSANT GRAPHIQUEMENT"
40 PRINT"LA LOI DU PESON A RESSORT"
50 PRINT:PRINT:PRINT TAB (10)"PROF.W.VOSS 1984":PRINT
55 PRINT:PRINT:PRINT:PRINT
60 INPUT "VALEUR DE LA CONSTANTE DU RESSORT  K:";K
70 CLS
75 LOCATE 3,3:PRINT"FORCE"
77 LOCATE 20,3:PRINT" K = ";K
80 PLOT 50,50:DRAW 380,50
90 PLOT 50,50:DRAW 50,350
100 LOCATE 25,22:PRINT"CM"
110 PLOT 50,50
120 B=K/10
125 A=380
130 E=50 + B*350
140 IF E>399 THEN A=400-(350*(E-399)/(E-49)):E=399
150 DRAW A,E
263 FOR I=1 TO 9999:NEXT I
265 CLS
270 INPUT "FORCE F                  :";F
280 FW=F/K
290 PRINT:PRINT:PRINT"CONSTANTE          :";K
300 PRINT:PRINT:PRINT"ALLONGEMENT        :";FW
310 PRINT:PRINT:PRINT:INPUT "UN AUTRE CALCUL           (O/N) ";A$
320 IF A$="O" THEN 265
330 PRINT:PRINT:PRINT:INPUT "REPRESENTATION GRAPHIQUE (O/N) ";A$
340 IF A$="O" THEN CLS : GOTO 60
350 PRINT:PRINT:PRINT"FIN":END
```

5) Liste des variables

A = colonne de fin du tracé de la droite correspondant à la constante
A$= Chaîne de caractères de réponse (Oui/Non)
B = K/10 (grandeur donnée pour la normalisation de l'affichage sur l'écran)
K = Constante du ressort
E = (idem A)
F = Force
FW= Allongement du ressort
I = Indice de boucle

6) Description du programme

Ligne 10-55 : titre
Ligne 60 : entrée de la constante du ressort K
Ligne 70-100 : visualisation du système de coordonnées
origine : 50,50
axe horizontal : 50,50 à 380,50
axe vertical : 50,50 à 50,350
désignations des axes (lignes 75 et 100) et indication de la constante (ligne 77)
Ligne 110-150 : tracé de la droite correspondant à la constante du ressort du point de coordonnées 50,50 au point A,E. Il faut modifier la valeur d'origine de A (A = 380, ligne 125) et la valeur de E si E devient supérieur à 399
Ligne 263-265 : pause et effacement de l'écran
Ligne 270 : demande d'une force F (poids)
Ligne 280 : calcul de l'allongement du ressort
Ligne 290-300 : sortie des résultats
Ligne 310 : demande à l'utilisateur s'il veut effectuer un autre calcul, si oui, retour à 265 si non, poursuite à la ligne 330

Ligne 330 : demande pour une nouvelle représentation graphique. Si oui, retour à la ligne 60 après effacement de l'écran si non, poursuite à la ligne 350

Ligne 350 : fin du programme

7) Résultats

Après avoir entré la constante du ressort, le programme construit un graphique semblable à celui présenté au début de cette section.

Après une pause et la donnée d'une force F, le programme calcule l'allongement du ressort correspondant. Un exemple de calcul est inutile ici (revoir à ce sujet l'exemple de la première étape).

4.5 Principe d'Archimède

1) Présentation du problème

Le principe d'Archimède s'énonce ainsi : Tout corps plongé dans un fluide subit une poussée égale au poids du fluide déplacé.

Le principe décrit le fait connu qu'il faut exercer une force non négligeable pour, par exemple, plonger dans l'eau un seau vide, le fond en bas.

Plus un corps est plongé dans un fluide, plus il semble perdre de son poids, ceci à cause de la pression qui s'exerce sur lui.

Le programme qui va maintenant être présenté devra déterminer, sur la base de ces relations, le volume et le poids (poids volumique) d'un corps de nom inconnu, plongé dans un fluide donné (l'eau, par exemple), connaissant son poids dans l'air.

2) Analyse du problème

Comme pour tous les énoncés de problèmes concrets il faut une fois encore procéder en tenant compte du schéma suivant :

E = Entrée
T = Traitement
S = Sortie

Cela signifie que nous devons d'abord déterminer les informations qui sont nécessaires à l'ordinateur pour résoudre le problème posé. Ces informations sont les suivantes :

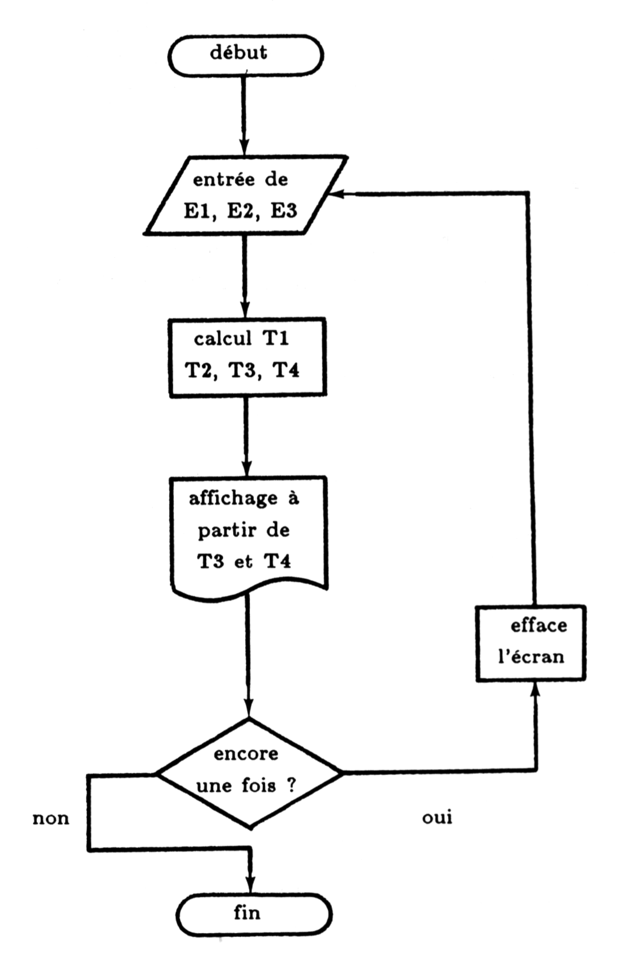
E1 - Poids dans l'air du corps dont on ignore le nom
E2 - Poids apparent du corps dans le fluide
E3 - Poids volumique du fluide

Une fois en possession de ces informations il est possible, au cours du traitement, de déterminer :

T1 - Poussée exercée par le fluide sur le corps et poids de la quantité de fluide déplacé
T2 - Volume de cette quantité de fluide
T3 - Volume du corps immergé (1er résultat)
T4 - Poids volumique du corps immergé (2ème résultat)

Les résultats obtenus dans les derniers point T3 et T4 sont à visualiser.

3) Organigramme



4) Programme

```
10 REM P2-ARCHIMEDE
20 CLS
30 PRINT"PROGRAMME POUR L'EMPLOI DU PRINCIPE"
40 PRINT"D'ARCHIMEDE":PRINT:PRINT
50 PRINT TAB(10)"PROF.W.VOSS 1984":PRINT
55 PRINT"AVEC LE PRINCIPE D'ARCHIMEDE, LE VOLUME ET"
56 PRINT"LE POIDS VOLUMIQUE"
57 PRINT"PEUVENT ETRE DETERMINES":PRINT:PRINT
70 PRINT"ENTREZ LES INFORMATIONS SUIVANTES"
80 PRINT:PRINT
90 PRINT TAB(5)"- POIDS DU CORFS DANS L'AIR"
100 PRINT TAB(5)"  (PA)"
110 PRINT:PRINT TAB(5)"- POIDS VIRTUEL DU CORPS"
120 PRINT TAB(5)"  DANS LE FLUIDE   (PV)"
130 PRINT:PRINT TAB(5)"- POIDS VOLUMIQUE DU FLUIDE  (SF)"
140 PRINT:PRINT:INPUT "  PA=";PA
150 PRINT:INPUT "  PV=";PV
160 PRINT:INPUT "  SF=";SF
170 PP=PA-PV
175 VF=PP/SF
180 VC=VF
190 SC=PA/VC
200 CLS
210 PRINT"CORPS A CONTROLER          : ":PRINT
220 PRINT:PRINT"     VOLUME           : ";VC
230 PRINT:PRINT"     POIDS VOLUMIQUE : ";SC
240 PRINT:PRINT:PRINT:INPUT "UN AUTRE CALCUL (O/N) ";A$
250 IF A$="O" THEN CLS:GOTO 70
260 PRINT:PRINT"FIN DES CALCULS":END
```

5) Liste des variables

A$ = variable chaîne de caractères de réponses Oui/Non
PA = poids du corps dans l'air
PV = poids virtuel du corps dans le fluide
PP = perte de poids apparente
VC = volume du corps
VF = volume du fluide déplacé
SF = poids volumique du fluide
SC = poids volumique du corps

6) Description du programme

Ligne 10-130 : titre, énoncé et informations explicatives
Ligne 140-160 : entrée des informations
Ligne 170-190 : étapes de traitement (calculs)
Ligne 200-230 : affichage des résultats
Ligne 240-260 : demande pour un autre exemple. Si oui, retour à la ligne 70 sinon fin du programme.

7) résultats

Entrons par exemple les informations suivantes :

E1 : poids du corps dans l'air = 70 g
E2 : poids virtuel du corps dans le fluide de contrôle (à mesurer avec un peson à ressort) = 40 g
E3 : poids volumique du fluide de contrôle = 1 (eau)

Le programme donne alors les résultats suivants :

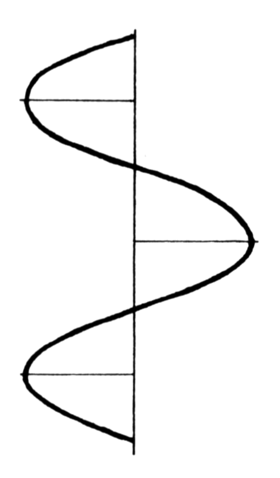
```
CORPS A CONTROLER :

VOLUME             = 30
POIDS SPECIFIQUE = 2.3333333
ENCORE UNE FOIS (O/N)?
```

4.6 Mouvement pendulaire

1) présentation du problème

Le mouvement pendulaire fait partie des principes fondamentaux en physique. Il joue un rôle très important dans l'étude du son (l'acoustique), de la lumière (l'optique) et de l'électricité.



La représentations de mouvements pendulaires est sinusoïdale comme le montre le schéma ci-contre.

Nous allons maintenant présenter un programme qui représente graphiquement l'oscillation sinusoïdale d'un pendule pour différentes amplitudes (A) et longueurs de période (P). A chaque période correspond ce que l'on appelle la fréquence (F) d'une oscillation.

2) Analyse du problème

Pour ce programme nous devons entrer l'amplitude de l'oscillation et la longueur de sa période.

Le programme, comme dans le schéma ci-dessus, représente à l'écran l'oscillation sinusoïdale verticalement (ce qui est plus simple à programmer qu'horizontalement).

Il consiste principalement à normaliser l'amplitude et la longueur de période de l'oscillation afin d'utiliser au mieux l'espace de l'écran.

On y parvient en positionnant la ligne médiane de l'oscillation en colonne 20 et en donnant à l'amplitude une étendue de 19 colonnes par exemple, ce qui correspond à la plus grande déviation possible à l'écran.

Nous n'avons donc pas choisi ici le mode graphique haute résolution (se reporter pour cela au chapitre 9), mais le mode normal par le biais d'un numéro de code ASCII approprié.

3) Organigramme

début
entrée A et P
normalisation de A et P
I = 1
calcul sinus
sortie
I = I + 1
terminé ?
oui
non
non

4) Programme

```
10 REM P3-PENDULE
20 CLS
30 PRINT"PROGRAMME DE SIMULATION"
40 PRINT"D'UN MOUVEMENT PENDULAIRE":PRINT:PRINT
50 PRINT:PRINT TAB(19) "PROF.W.VOSS 1984":PRINT
52 PRINT:PRINT"APPUYEZ SUR <ESC> POUR INTERROMPRE"
53 PRINT"LE DEROULEMENT DU PROGRAMME"
54 PRINT:PRINT"APPUYEZ DEUX FOIS SUR <ESC> POUR"
56 PRINT"ARRETER LE PROGRAMME"
60 PRINT:INPUT "AMPLITUDE DE L'OSCILLATION    :";A
70 PRINT:INPUT "LONGUEUR DE LA PERIODE        :";P
75 IF A>19 THEN A=19
80 CLS
90 I=1
100 J=(2*(22/7)/P)*I
110 Y=SIN (J)
120 S=20+Y*A
130 FOR K=1 TO S
140 PRINT CHR$(231);
150 NEXT K
160 PRINT
170 I=I+1: GOTO 100
```

5) Liste des variables

A = amplitude de l'oscillation
I = indice de boucle
J = calcul de la variable : J = (2 * pi / p) *I
(J sert à la normalisation de la longueur de la période)
K = indice de boucle (compteur) déterminant, à l'aide de la valeur M, le nombre de points à visualiser.
P = longueur de la période de l'oscillation
M = dernière colonne de la ligne en cours
Y = SIN (J)

6) Description du programme

Ligne 10-55 : titre et explications
Ligne 60-70 : demande des informations à entrer
Ligne 75 : limitation de l'amplitude de l'oscillation à la valeur maximum 19
Ligne 80 : effacement de l'écran
Ligne 90 : initialisation de l'indice (I) du compteur de lignes
Ligne 100 : normalisation de la longueur de la période P dans la variable J
Ligne 110 : détermination de la valeur sinusoïdale
Ligne 120 : détermination de la fin de la ligne pour la valeur sinusoïdale du moment en tenant compte du fait que la valeur zéro doit se trouver dans la 20ème colonne
Ligne 130-150 : boucle destinée à remplir de "petites boules" la ligne jusqu'à la valeur sinusoïdale du moment.
Ligne 160-170 : saut de ligne et visualisation de la ligne suivante par le retour à la ligne 100 après incrémentation de l'indice I de 1.

7) Résultats

Les résultats obtenus correspondent au schéma présenté dans la première étape à la différence près que l'espace se trouvant entre le bord gauche de l'écran et la ligne sinusoïdale est comblé par de "petites boules" jaunes.

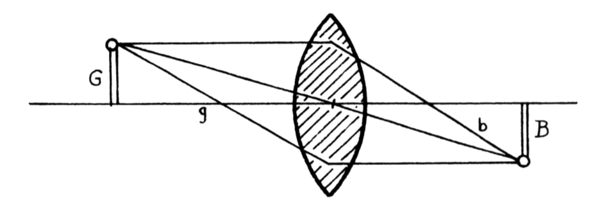
Le lecteur peut facilement imaginer comment modifier le programme pour supprimer ce "remplissage", il lui suffit de ne visualiser (par de "petites boules" par exemple), que les positions correspondant à la ligne sinusoïdale.

Le programme présenté ici est un programme sans fin qui ne peut être interrompu qu'en frappant la touche ESC. Ceci peut être également facilement modifié.

4.7 Représentation optique

1) Présentation optique

Une lentille convexe permet comme le montre de manière schématique la représentation suivante de donner une image d'un objet quelconque :



L'appareil photographique et l'oeil humain fonctionnent de la même manière que cette lentille.

Nous allons maintenant présenter un programme qui détermine la taille ou la proportion de l'image en fonction de la grandeur de l'objet à saisir, la distance la séparant de la lentille et la distance focale de la lentille convexe.

2) Analyse du problème

Les informations nécessaires au programme sont :

1 taille des objets
2 distance séparant l'objet de la lentille
3 distance focale de la lentille

Pour résoudre ce problème nous devons connaître l'équation de la lentille :

1/f = 1/g + 1/b

avec

F = distance focale
g = distance de l'objet à la lentille
b = distance de l'image à la lentille

Pour connaître b, nous devons transformer cette équation en :

b = 1 / (1/f - 1/g)

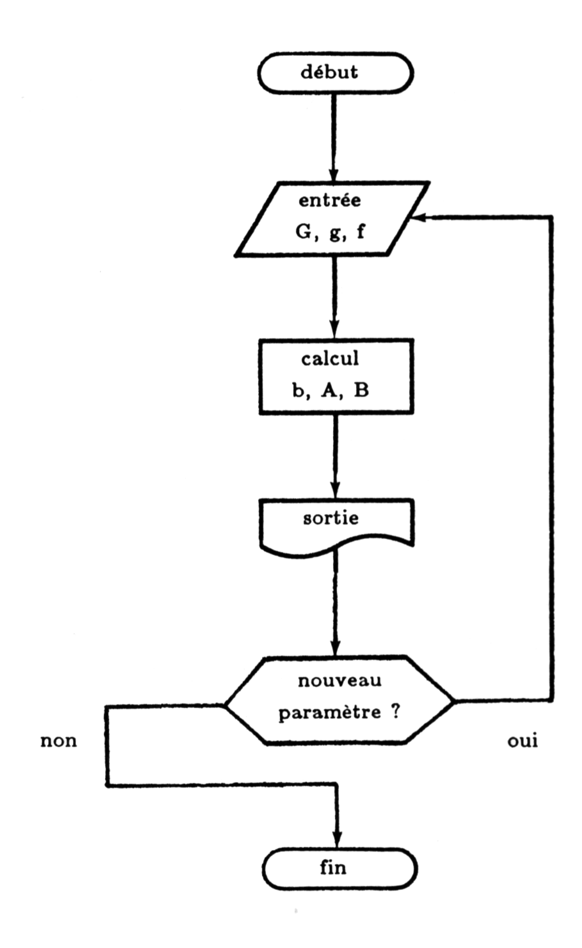
De plus on peut dire que la proportion A de l'image par rapport à l'objet satisfait à la relation :

A = b/g = B/G

Si l'objet à représenter a par exemple la taille T = 20 cm la taille de son image peut être déterminée à l'aide de cette relation, et ainsi :

B = (b * G)/g, sachant que B/G = b/g

3) Organigramme



4) Programme

```
10 REM P4-LENTILLE
20 CLS
30 PRINT"PROGRAMME DESTINE A CALCULER LA DISTANCE D'UNE IMAGE"
40 PRINT"TAILLE DE L'IMAGE ET PROPORTION DE LA REPRESENTATION"
50 PRINT"AVEC UNE LENTILLE CONVEXE"
60 PRINT TAB(10)"PROF.W.VOSS 1984":PRINT
70 PRINT"ENTREZ LES DONNEES":PRINT
80 PRINT TAB(5)"TAILLE DE L'OBJET  :";:INPUT G
90 PRINT TAB(5)"DISTANCE DE L'OBJET:";:INPUT GW
100 PRINT TAB(5)"DISTANCE FOCALE    :";:INPUT F
110 CLS
120 BW=1/(1/F-1/GW)
130 A=BW/GW
140 B=(G*BW)/GW
145 BW=INT(BW*100+0.5)/100:A=INT(A*100+0.5)/100:B=INT(B*100+0.5)/100
150 PRINT"DONNEES   :":PRINT
160 PRINT TAB(5)"DISTANCE FOCALE            :";F
170 PRINT:PRINT TAB(5)"TAILLE DE L'OBJET          :";G
180 PRINT:PRINT TAB(5)"DISTANCE DE L'OBJET        :";GW
190 PRINT:PRINT:PRINT"RESULTATS :":PRINT
200 PRINT:PRINT TAB(5)"PROPORTION DE LA REPRESENTATION :";A
210 PRINT:PRINT TAB(5)"TAILLE DE L'IMAGE               :";B
220 PRINT:PRINT TAB(5)"DISTANCE DE L'IMAGE             :";BW
230 PRINT:PRINT:PRINT:INPUT "ENCORE UN CALCUL (O/N) ";A$
240 IF A$="O" THEN CLS:GOTO 70
250 PRINT:PRINT"FIN DES CALCULS":END
```

5) Liste des variables

A = proportion de la représentation
A$ = variable chaîne de caractères (Oui/Non)
B = taille de l'image à la lentille
BW = distance de l'image
F = distance focale
G = taille de l'objet
GW = distance de l'objet à la lentille

6) Description du programme

Ligne 10-70 : titre et explications
Ligne 80-100 : demande des informations à entrer
Ligne 110-140 : effacement de l'écran et détermination de BW et B
Ligne 145 : normalisation à 2 décimales de BW, A et B
Ligne 150-220 : sortie des résultats
Ligne 230-240 : demande à l'utilisateur s'il veut à nouveau un calcul (si oui retour à la ligne 70 après effacement de l'écran)
Ligne 250 : fin du programme

7) Résultats

Entrons par exemple : G = 20 cm, GW = 300 cm, F = 15 cm, nous obtenons :

```
TAILLE DE L'OBJET    : 20
DISTANCE DE L'OBJET  : 300
DISTANCE FOCALE      : 15
```

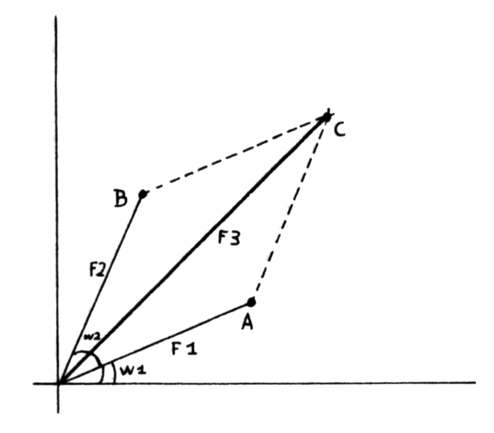
RESULTATS :

PROPORTION DE LA REPRESENTATION: .05
TAILLE DE L'IMAGE : 1.05
DISTANCE DE L'IMAGE : 15.79

ENCORE UN CALCUL (O/N) ?

4.8 Parallélogramme des forces

Quand deux forces s'appliquent à un même point il en résulte une force qui peut être représentée par une diagonale dans le parallélogramme des forces :



Dans le programme qui suit nous représentons graphiquement le parallélogramme correspondant à 2 forces quelconques. Calculons la force résultante (F3).

2) Analyse du problème

Le point d'application de ces deux forces doit se situer au centre de l'écran (320, 200).

Les informations à saisir correspondant aux forces F1 et F2 doivent tout d'abord être converties de manière à ce que F1 + F2 ne soit pas supérieur à 320 ceci pour que la représentation à l'écran de la résultante F3 soit possible.

Il faut encore entrer les angles W1 et W2 que font respectivement les forces F1 et F2.

Le théorème de Pythagore et les relations sur les angles dans le triangle rectangle permettent ensuite de calculer sur quelles colonnes et quelles lignes se trouvent les points A et B et d'en déduire les coordonnées du point C.

Puis connaissant la position du point C, on peut, à partir de F1 et de F2 calculer la force F3 qui nous intéresse.

3) Organigramme

Il n'y a pas d'organigramme ici.

4) Programme

```
10 REM P5-PARALLELOGRAMME
20 CLS
30 PRINT"PROGRAMME ILLUSTRANT LE PARALLELOGRAMME"
40 PRINT"DES FORCES DANS LE CAS OU DEUX FORCES"
50 PRINT TAB(9)"S'AFFRONTENT"
60 PRINT:PRINT TAB(10)"PROF.W.VOSS 1984":PRINT
70 SU=320:ZU=200
80 PRINT:PRINT:PRINT:PRINT
90 INPUT "PREMIERE FORCE   F1:";F1
100 INPUT "ANGLE             F1:";W1
105 PRINT:PRINT
110 INPUT "DEUXIEME FORCE   F2:";F2
120 INPUT "ANGLE             F2:";W2
130 W1=W1*((22/7)/180):W2=W2*((22/7)/180)
135 H=F1:G=F2
140 F1=(F1/(G+F1))*300:F2=(F2/(H+F2))*300
150 Z1=F1*SIN(W1)+200
160 S1=((Z1-200)/TAN(W1))+320
165 CLS
170 PLOT 320,200
180 DRAW S1,Z1
190 Z2=F2*SIN(W2)+200
200 S2=((Z2-200)/TAN(W2))+320
210 PLOT 320,200
220 DRAW S2,Z2
230 S3=S2+(S1-SU)
240 Z3=Z2+(Z1-ZU)
250 PLOT SU,ZU
260 DRAW S3,Z3
270 F3=SQR((Z3-ZU)^2+(S3-ZU)^2)
275 F3=F3*(G+H)/300
280 LOCATE 5,16
290 PRINT"FORCE 1 = ";H
300 PRINT TAB (5)"FORCE 2 = ";G
310 PRINT:PRINT"FORCE RESULTANTE = ";F3
```

```
320 PRINT:PRINT:PRINT:INPUT "ENCORE UN CALCUL  (O/N) ";A$
330 IF A$="O" THEN CLS:GOTO 90
340 PRINT:PRINT:PRINT"FIN":END
```

5) Liste des variables

A$ = variable chaîne de caractère de réponse (Oui/Non)
F1 = première force
F2 = deuxième force
F3 = troisième force (résultante)
G = grandeur secondaire (=F2)
H = grandeur secondaire (=F1)
SU = colonne origine
S1 = colonne point A
S2 = colonne point B
S3 = colonne point C
W1 = angle de la force F1 avec l'horizontale
W2 = angle de la force F2 avec l'horizontale
ZU = ligne origine
Z1 = ligne point A
Z2 = ligne point B
Z3 = ligne point C

6) Description du programme

Ligne 10-60 : titre et explications
Ligne 70 : coordonnées d'origine
Ligne 80-120 : entrée des informations
Ligne 130 : conversion de l'angle en radian
Ligne 135 : stockage des valeurs des forces F1 et F2 dans les variables H et G
Ligne 140 : normalisation des deux forces pour une bonne utilisation de l'écran
Ligne 150-160 : détermination des coordonnées de A
Ligne 165-180 : tracé de F1

Ligne 190-200 : coordonnées de B
Ligne 210-220 : tracé de F2
Ligne 230-240 : coordonnées de C
Ligne 250-260 : tracé de F3
Ligne 270 : calcul de F3 (Pythagore)
Ligne 275 : annulation de la normalisation de la ligne 140 pour F3
Ligne 280-310 : sortie du résultats
Ligne 320-340 : fin du programme, à moins qu'un autre calcul ne soit souhaité (si oui retour à la ligne 90)

7) Résultats

Voir la représentation figurant au paragraphe 1).

4.9 La loi d'Ohm

1) Présentation du problème

La loi d'Ohm s'applique aux conducteurs pour lesquels l'intensité augmente proportionnellement à la tension.

La formule est la suivante :

tension / intensité = constante = R

On appelle R la résistance électrique qu'oppose un conducteur.

Dans ce programme nous nous proposons, après entrée de deux valeurs (tension, intensité ou résistance), de calculer la troisième de ces valeurs.

Cela nous donne l'occasion d'introduire une nouvelle instruction, **PRINT USING** :

La ligne

X = 12.323 : PRINT USING" . ",X

a pour résultat 12.32.

Le "masque" " " (ou autres) permet d'indiquer le nombre de chiffres avec lequel un résultat doit apparaître.

2) Analyse du problème

Le programme traité ne présente aucune difficulté sur le plan mathématique, un simple calcul étant suffisant.

Le seul point délicat est ici la logique de programmation.
Trois combinaisons des données sont possibles, chacune ayant sa propre procédure.

Les données peuvent être :

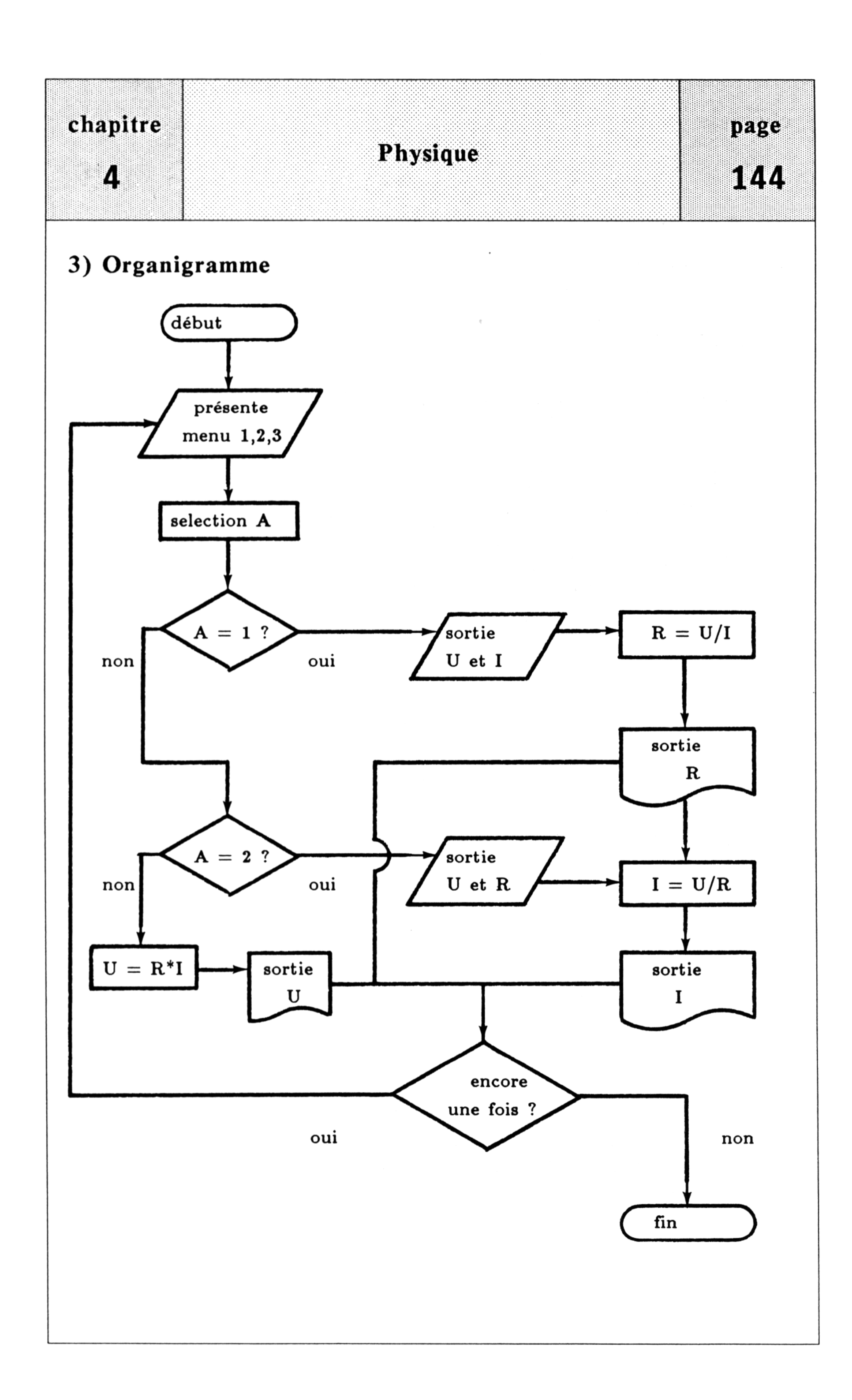
1. tension et intensité,
2. tension et résistance,
3. intensité et résistance.

Ce qui nous permet de présenter la notion de menu :

Il s'agit de présenter à l'utilisateur les différentes possibilités à l'écran, sous forme de choix. Celui-ci doit alors faire sa sélection. Selon son choix, différentes procédures de calcul, sont lancées.

Le menu ici propose un choix de trois possibilités (voir plus haut).

3) Organigramme



4) Programme

```
10 REM P6-LOI D'OHM
20 CLS
30 PRINT"  PROGRAMME REPRESENTANT LA RELATION"
40 PRINT"ENTRE INTENSITE, TENSION ET RESISTANCE"
50 PRINT TAB(14)"LOI D'OHM"
60 PRINT:PRINT:PRINT TAB(9)"PROF.W.VOSS 1984":PRINT
70 PRINT:PRINT"AVEC QUELLE PRESENTATION TRAVAILLE-T-ON ?":PRINT
80 PRINT TAB(5) "TENSION ET INTENSITE          (1)"
90 PRINT:PRINT TAB(5)"TENSION ET RESISTANCE         (2)"
100 PRINT:PRINT TAB(5)"INTENSITE ET RESISTANCE      (3)"
110 PRINT:PRINT:INPUT "ENTREZ VOTRE CHOIX S.V.P ";A
120 CLS
130 IF A>1 THEN 190
140 INPUT "TENSION      : ";U
150 INPUT "INTENSITE    : ";I
160 R=U/I
170 PRINT:PRINT:PRINT"RESISTANCE   : ";USING "#####.##";R
180 GOTO 300
190 IF A=3 THEN 250
200 INPUT "TENSION      : ";U
210 INPUT "RESISTANCE   : ";R
220 I=U/R
230 PRINT:PRINT:PRINT"INTENSITE    : ";USING "#####.##";I
240 GOTO 300
250 INPUT "RESISTANCE   : ";R
260 INPUT "INTENSITE    : ";I
270 U=R*I
280 PRINT:PRINT:PRINT"TENSION      : ";USING "#####.##";U
300 PRINT:PRINT:PRINT:INPUT "ENCORE UN CALCUL (O/N) ";A$
310 IF A$="O" THEN CLS: GOTO 70
320 PRINT:PRINT"FIN DES CALCULS":END
```

5) Liste des variables

A = variable de sélection (menu)
A$= variable chaîne de caractère de réponse (oui/non)
I = intensité
R = résistance
U = tension

6) Description du programme

Ligne 10-60 : titre et explications
Ligne 70-100 : présentation du menu
Ligne 110 : sélection du menu
Ligne 120-130 : Si A est supérieur à 1 on continue à la ligne 190 après avoir effacé l'écran.
Ligne 140-150 : entrée des informations en fonction de la sélection A = 1
Ligne 160-170 : calculs et sortie du résultat
Ligne 180 : saut à la ligne 300
Ligne 190-240 : calculs correspondants au deuxième choix A = 2
Ligne 250-280 : calculs correspondants au troisième choix A = 3
Ligne 300-310 : interrogation pour savoir si l'utilisateur désire à nouveau un calcul. Si oui, effacement de l'écran et retour à la ligne 70
Ligne 320 : fin du programme

7) Résultats

Choisissons par exemple parmi les choix du menu, le chiffre 1. Le programme nous demande alors la tension et l'intensité. Entrons, par exemple, la valeur 220 pour la tension et 11 pour l'intensité. Le résultat est le suivant :

RESISTANCE : 20
ENCORE UN CALCUL (O/N) ?

LANGUES



5.1 Avant-propos, compléments BASIC

L'ordinateur peut également nous apporter une aide efficace dans le domaine des langues étrangères. Il ne s'agit pas là de présenter des programmes de cours de langue (pour apprendre l'anglais de A à Z par exemple), ceci dépasserait le but, plus modeste, que nous nous sommes fixé dans ce livre. Il faut cependant savoir qu'il est possible d'acquérir ces programmes complets pour un prix relativement raisonnable.

On se penchera plutôt sur des programmes simples qui pourront seconder l'élève dans l'étude de la langue qu'il a choisie.

Il sera question, avant tout, des programmes qui facilitent l'apprentissage du vocabulaire ou la préparation de traductions. Ils apportent une aide efficace car ils évitent d'avoir constamment recours aux dictionnaires.

Ces programmes exigent cependant la connaissance de quelques instructions supplémentaires propres au langage BASIC :

Instruction 20 :

```
nn    nom$ = INKEY$
```

Cette instruction, tout comme INPUT attend l'entrée d'une donnée qui est ensuite attribuée à une variable string (nom$).

Il n'est pas nécessaire de valider cette entrée par la touche ENTER. De plus amples détails sur l'utilisation de cette instruction seront fournis dans les programmes à venir.

Il est souvent utile de regrouper des instructions en sous parties pour les rappeler plusieurs fois à partir du programme principal.

On appelle sous-programmes (subroutines) ces programmes partiels. Le saut, du programme principal vers le sous-programme ne peut se faire qu'à l'aide de l'instruction suivante :

Instruction 21 :

nn GOSUB mm

Cette instruction opère un saut dans le sous-programme dont la 1ère ligne a le numéro mm. Une fois que ce sous-programme est exécuté, il faut effectuer un saut en arrière dans le programme principal.

Pour cela la dernière instruction du sous-programme doit être la suivante :

Instruction 22 :

nn RETURN

Cette instruction permet de revenir directement à ce qui suit l'instruction GOSUB dans le programme principal.

Ces instructions n'ont été définies que d'un point de vue théorique. Leurs procédés d'utilisation seront explicités dans des programmes ultérieurs.

5.2 Verbes irréguliers anglais

1) Présentation du problème

Dans un premier exemple, nous allons présenter un programme qui indique les formes d'un verbe anglais irrégulier.

A la saisie, par exemple, du verbe :

ALLER (go)

doivent s'afficher à l'écran :

GO WENT GONE

Par souci de simplification de la programmation nous nous limiterons aux 20 verbes irréguliers les plus usités. Le lecteur pourra facilement extrapoler pour compléter le programme afin d'obtenir la liste complète des verbes irréguliers anglais.

2) Analyse du problème

Après l'entrée d'un verbe, le programme doit donner les trois formes et la traduction française.

Ceci implique que ces informations doivent déjà être stockées dans l'ordinateur.

Si nous disposons d'un ensemble de données important, il n'est pas utile de recréer, à chaque lancement du programme, la base d'informations. Il est plus judicieux à chaque fois de n'entrer en mode conversationnel, que les nouvelles données qui seront stockées dans un fichier spécialement créé sur une disquette.

Il n'est pas dans notre propos ici d'examiner cette possibilité qui demande à l'utilisateur une certaine maîtrise de l'emploi des disquettes.

L'ordinateur recherche ensuite ce verbe parmi les informations de sa base de données jusqu'à ce qu'il l'ait trouvé. Il peut alors afficher ses formes et sa traduction française.

Cependant, si la recherche est infructueuse, deux explications sont possibles :

- soit ce verbe ne figure pas dans l'ensemble des données (nous n'avons pour l'instant qu'une base de 20 verbes),

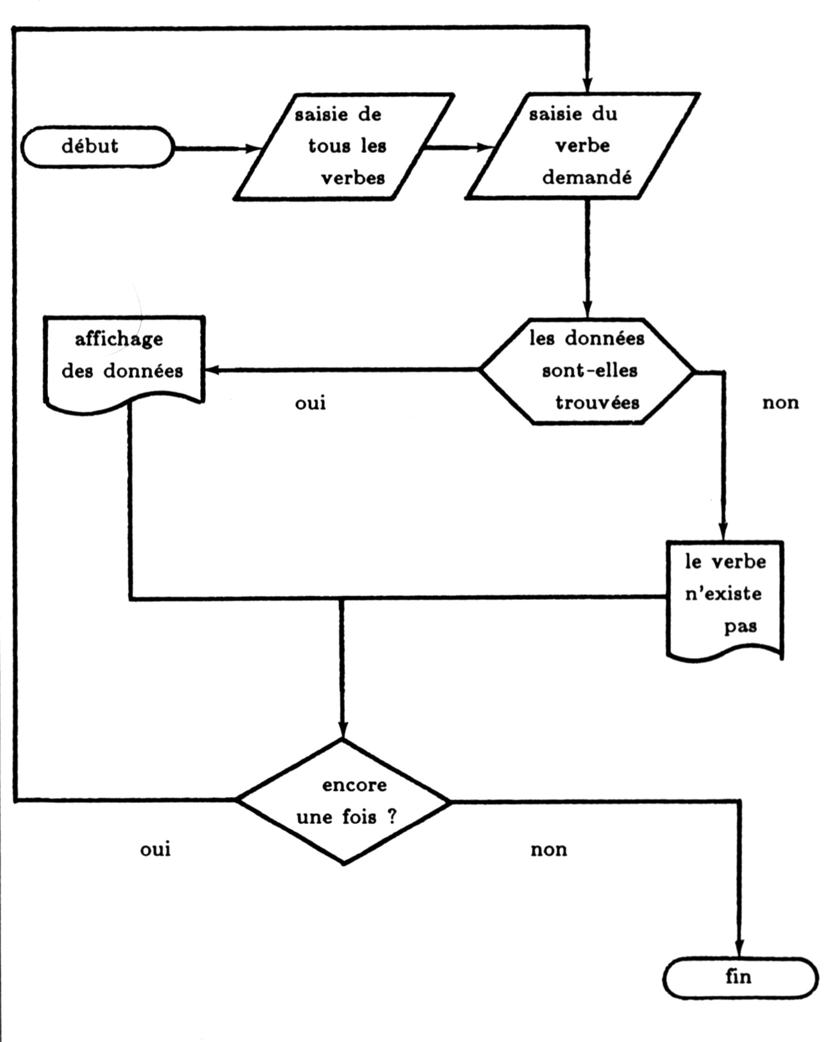
- soit ce verbe n'est pas irrégulier.

Dans les deux cas le même message apparaît à l'écran.

Ce programme permet également à l'utilisateur de proposer un autre verbe.

Les formes et la traduction française peuvent être stockées dans un tableau de chaîne de caractères à deux dimensions.

3) Organigramme



4) Programme

```
10 REM S1-VERBES IRREGULIERS ANGLAIS
20 CLS
30 PRINT"PROGRAMME DESTINE A SORTIR LES RADICAUX"
40 PRINT"DES VERBES IRREGULIERS ANGLAIS ENTRES"
50 PRINT
60 PRINT:PRINT:PRINT TAB(10) "PROF.W.VOSS 1984":PRINT
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"LE NOMBRE DES MOTS PRESENTS EST"
90 PRINT"LIMITE DANS CE PROGRAMME. DES"
100 PRINT"ELARGISSEMENTS SONT POSSIBLES SI"
110 PRINT"LA LISTE DES DATAS EST ALLONGEE"
120 PRINT"ET SI LA VALEUR DONNE A 'N' A LA"
125 PRINT"LIGNE 160 EST AUGMENTEE D'AUTANT"
130 LOCATE 5,23:PRINT"PRESSEZ UNE TOUCHE S.V.P"
140 A$=INKEY$
150 IF A$="" THEN 140
155 CLS:PRINT TAB(12) "UN MOMENT S.V.P":PRINT:PRINT
160 N=20
170 DIM W$(N,4)
180 FOR I=1 TO N:FOR J=1 TO 4:READ W$(I,J):NEXT J:NEXT I
190 CLS
200 PRINT"ENTREZ LE VERBE FRANCAIS"
210 PRINT"DONT VOUS VOULEZ CONNAITRE"
220 INPUT "LES RADICAUX  : ";V$
230 FOR I=1 TO N
240 IF V$=W$(I,1) THEN 275
250 NEXT I
260 PRINT:PRINT:PRINT"VERBE NON PRESENT DANS LE STOCK DES DONNEES"
270 PRINT"OU PAS IRREGULIER ":GOTO 310
275 PRINT:PRINT:PRINT
280 PRINT"VERBE : ";W$(I,1):PRINT:PRINT:PRINT
290 PRINT"RADICAUX ANGLAIS  : ":PRINT
300 FOR J=2 TO 4:PRINT W$(I,J);"  ";:NEXT J
310 PRINT:PRINT:PRINT:INPUT "ENCORE UNE FOIS (O/N) ";A$
320 IF A$="O" THEN CLS:GOTO 200
```

```
330 PRINT:PRINT:PRINT"FIN":END
500 DATA ETRE,BE,WAS,BEEN
502 DATA AVOIR,HAVE,HAD,HAD
504 DATA COMMENCER,BEGIN,BEGAN,BEGUN
506 DATA APPORTER,BRING,BROUGHT,BROUGHT
508 DATA VENIR,COME,CAME,COME
510 DATA FAIRE,DO,DID,DONE
512 DATA MANGER,EAT,ATE,EATEN
514 DATA ACHETER,BUY,BOUGHT,BOUGHT
516 DATA SENTIR,FEEL,FELT,FELT
518 DATA TROUVER,FIND,FOUND,FOUND
520 DATA ALLER,GO,WENT,GONE
522 DATA PERDRE,LOSE,LOST,LOST
524 DATA FABRIQUER,MAKE,MADE,MADE
526 DATA LIRE,READ,READ,READ
528 DATA COURIR,RUN,RAN,RUN
530 DATA DIRE,SAY,SAID,SAID
532 DATA VOIR,SEE,SAW,SEEN
534 DATA S'ASSOIR,SIT,SAT,SAT
536 DATA PARLER,SPEAK,SPOKE,SPOKEN
538 DATA PRENDRE,TAKE,TOOK,TAKEN
```

5) Liste des variables

A$= variable chaîne de caractère de réponse
I = indice de boucle (compteur)
J = indice de boucle

N = nombre de données en stock
V$= verbe demandé
W$= formes du verbe et traduction française
(ensemble de données)

6) Description d'un programme

Ligne 10-135 : titre, explications et remarques sur la manière dont l'ensemble de données peut être mis à jour.
Ligne 140-150 : interruption du programme. L'exécution du programme ne continue que si l'utilisateur frappe n'importe quelle touche
Ligne 155 : effacement de l'écran et message
Ligne 160-180 : le tableau des données W$ est dimentionné puis initialisé par la lecture des données
Ligne 190-220 : effacement de l'écran et demande du verbe à rechercher (stocké dans V$)
Ligne 230-250 : recherche dans l'ensemble de données du verbe saisi. S'il s'y trouve, saut à la ligne 275 sinon, passage à la ligne 260
Ligne 260-270 : message à l'écran pour indiquer que le verbe est absent de l'ensemble de données, puis passage à la ligne 310
Ligne 275-300 : affichage des informations recherchées
Ligne 310-320 : interrogation pour savoir si une autre recherche est désirée. Si oui, effacement de l'écran et retour à la ligne 200 sinon, poursuite à la ligne 330
Ligne 330 : fin du programme
Ligne 500-538 : données

7) Résultats

Si, en réponse à la demande de la ligne 200, on entre :

ALLER

l'ordinateur répond :

VERBE ALLER
FORMES ANGLAISES :
GO WENT GONE
ENCORE UNE FOIS (O/N) ?

Si la liste des verbes irréguliers doit être complétée on modifiera le programme en ajoutant de nouvelles données (DATA) à partir de la ligne 500. Il faut alors modifier la ligne 160 en remplacant le chiffre 20 par le nouveau nombre total de verbes stockés.

5.3 Traduction de vocabulaire

1) Présentation du problème

Nous allons montrer comment l'ordinateur peut se substituer à un répertoire de vocabulaire. Il répond à un mot entré en français par le terme allemand correspondant ou vice versa (on peut faire ici la même remarque que dans la section 5.2 au sujet de l'utilisation de disquette en mode conversationnel).

Pour les mêmes raisons que précédamment, nous nous limitons à 10 mots. Ce programme n'est donné qu'à titre d'exemple.

S'il devait être utilisé véritablement, il devrait être pourvu d'un ensemble de donnéés plus important.

Ceci dit, le principe de fonctionnement de ce programme reste le même dans les deux cas.

2) Analyse du problème

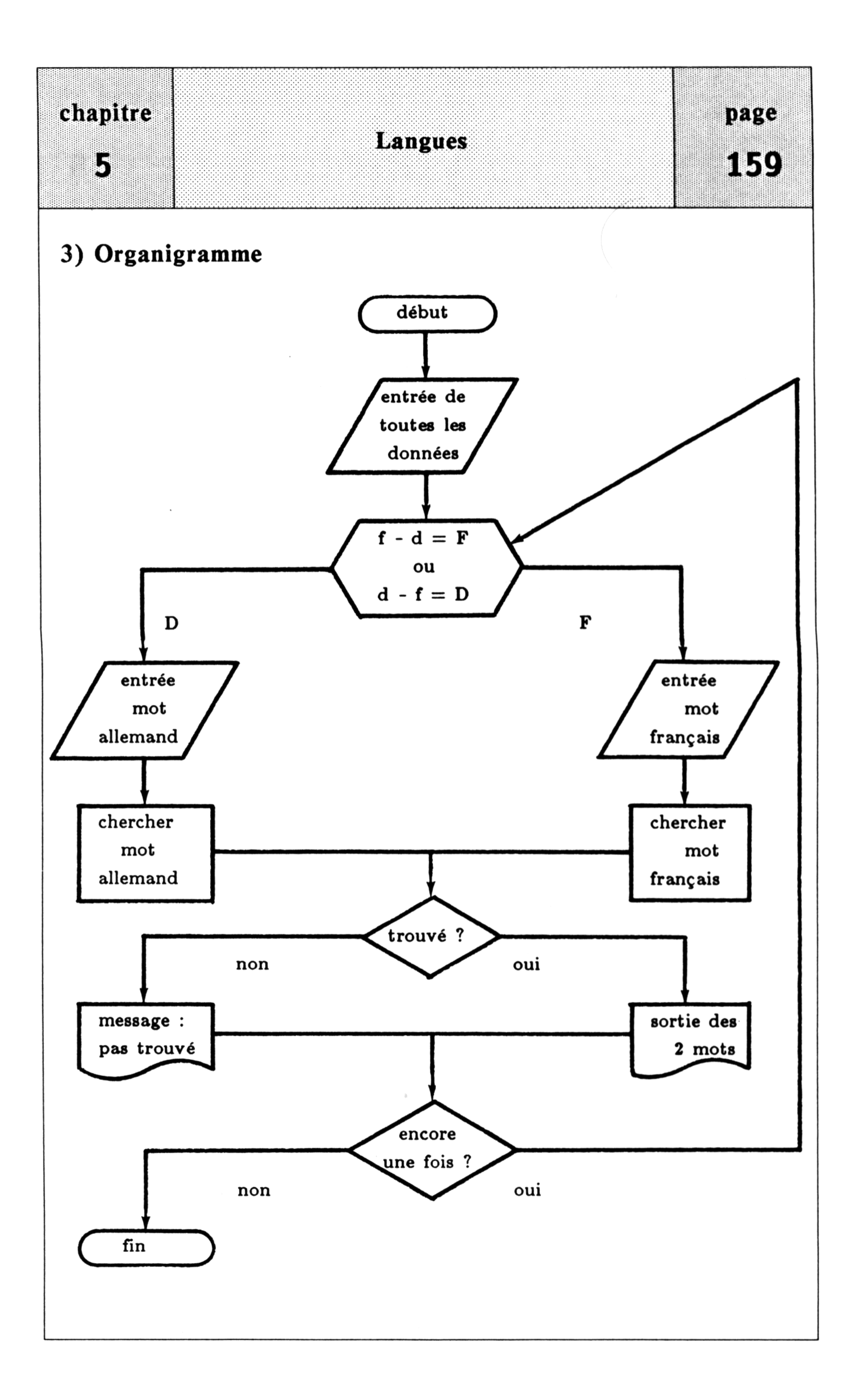
Lorsque l'utilisateur entre un mot français à l'écran, le programme effectue sa recherche dans l'ensemble de données en le comparant à sa liste de vocabulaire français (chaque mot français est associé à sa traduction allemande sous forme de paire). On procède de manière analogue pour obtenir la traduction française d'un terme allemand.

Si l'on veut, dans le même programme obtenir des traductions de type thème et de type version il faut commencer par indiquer à l'ordinateur le sens de traduction désiré.

Si le mot demandé ne peut pas être traduit parce qu'il ne figure pas dans la liste du vocabulaire (très courte ici), le programme affiche à nouveau un message.

De plus, l'utilisateur peut obtenir la traduction d'un autre mot s'il le désire.

3) Organigramme



4) Programme

```
10 REM S2-VERBES FRANCAIS-ALLEMAND
20 CLS
30 PRINT"PROGRAMME POUR CONSULTER DES VOCABLES:"
40 PRINT:PRINT"FRANCAIS-ALLEMANDS OU VICE VERSA"
50 PRINT:PRINT:PRINT TAB (10) "PROF.W.VOSS 1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT"DANS LE STOCK DE DONNEES DE CE PROGRAMME"
80 PRINT"IL N'Y A QUE DIX VOCABLES":PRINT
90 PRINT"SI LE PROGRAMME DOIT ETRE AGRANDI, IL FAUT:"
100 PRINT"AJOUTER A LA SUITE DE LA LIGNE 518,"
110 PRINT"D'AUTRES DONNEES ET MODIFIER LA VALEUR"
120 PRINT"'N' A LA LIGNE 160"
130 LOCATE 5,23:PRINT"PRESSEZ UNE TOUCHE S.V.P"
140 A$=INKEY$:IF A$="" THEN 140
150 CLS:PRINT"UN MOMENT S.V.P"
160 N=10
170 DIM D$(N),F$(N)
180 FOR I=1 TO N:READ F$(I),D$(I):NEXT I
190 CLS
200 PRINT"QUELLE EST LA TRADUCTION SOUHAITEE ?"
210 PRINT:PRINT:PRINT"-FRANCAIS : ALLEMAND   (1)"
220 PRINT:PRINT"OU"
230 PRINT:PRINT"-ALLEMAND : FRANCAIS   (2)"
240 PRINT:PRINT:INPUT" ENTREZ   1 OU 2";Z
245 CLS
250 IF Z=2 THEN 350
260 PRINT:PRINT:INPUT "MOT FRANCAIS S.V.P";W$
280 FOR I=1 TO N:IF W$=F$(I) THEN 320
290 NEXT I
300 PRINT:PRINT"MOT RECHERCHE ABSENT DES DONNEES"
310 PRINT:GOTO 450
320 CLS:PRINT"FRANCAIS     : ";F$(I)
330 PRINT:PRINT"ALLEMAND     : ";D$(I)
340 GOTO 450
350 PRINT:PRINT:INPUT "MOT ALLEMAND S.V.P : ";W$
```

```
360 FOR I=1 TO N:IF W$=D$(I) THEN 400
370 NEXT I
380 PRINT:PRINT"MOT RECHERCHE ABSENT DES DONNEES"
390 PRINT:GOTO 450
400 CLS:PRINT"ALLEMAND     : ";D$(I)
410 PRINT:PRINT"FRANCAIS     : ";F$(I)
450 PRINT:PRINT:PRINT:INPUT "ENCORE UNE FOIS  (O/N) ";A$
460 IF A$="O" THEN CLS:GOTO 200
470 PRINT:PRINT"FIN":END
500 DATA ACHETER,KAUFEN
502 DATA AVOIR,HABEN
504 DATA ALLER,GEHEN
506 DATA VOIR,SEHEN
508 DATA BOIRE,TRINKEN
510 DATA FAIRE,MACHEN
512 DATA LIRE,LESEN
514 DATA MANGER,ESSEN
516 DATA PRENDRE,NEHMEN
518 DATA FUMER,RAUCHEN
```

5) Liste des variables

A$= variable chaîne de caractère d'entrée

F$ = mots français
D$= mots allemands

I = indice de boucle (compteur)
N = nombre de données à stocker

W$= mot à traduire

Z = indice de sélection du menu

6) Description du programme

Ligne 10-120 : titre, explications, etc
Ligne 130-150 : attente, jusqu'à la pression d'une touche et effacement de l'écran
Ligne 160-180 : indication du nombre de données à stocker, les tableaux de données D$ et F$ sont dimensionnés puis remplis par lecture des données
Ligne 190-240 : effacement de l'écran et affichage du menu
Ligne 245-250 : effacement de l'écran si 2 est entré (traduction allemand-français), poursuite à la ligne 350
Ligne 260 : entrée d'un mot français
Ligne 280-290 : recherche des données correspondantes
Ligne 300-310 : si le mot recherché n'est pas trouvé poursuite à la ligne 450
Ligne 320-330 : visualisation des résultats
Ligne 340 : saut à la ligne 450
Ligne 350-410 : démarche similaire pour l'entrée d'un mot allemand
Ligne 450-470 : possibilité de répétition et fin du programme
Ligne 500-518 : données

7) Résultats

Une fois le menu à l'écran, frappons le chiffre 1 (traduction français-allemand). L'ordinateur nous demande alors un mot français.

Entrons, par exemple, le mot avoir, l'ordinateur répond alors :

```
FRANCAIS : AVOIR
ALLEMAND : HABEN

ENCORE UNE FOIS (O/N) ?
```

5.4 Test de vocabulaire anglais

1) Présentation du problème

Comme dans la section précédente il s'agit à partir d'un stock de vocabulaire d'effectuer des traductions. Mais ici, l'utilisateur doit donner la traduction anglaise exacte du mot en français choisi au hasard par l'ordinateur.

Ce programme fonctionne comme le précédent à la différence près qu'ici le carnet de vocabulaire est sous forme d'un tableau à 2 dimensions (stockage des mots anglais sur une seule colonne).

Bien entendu, ce ne sont ni le choix de la langue ni le sens de traduction (version ou thème) qui modifient l'allure générale du programme.

Toujours par souci de simplification nous présentons, à titre d'exemple, un programme qui n'utilise que dix mots de vocabulaire.

L'ordinateur n'en choisit au hasard qu'une partie (appelée sous groupe) et affiche la côte de réussite de l'utilisateur à la fin de chaque exercice.

2) Analyse du problème

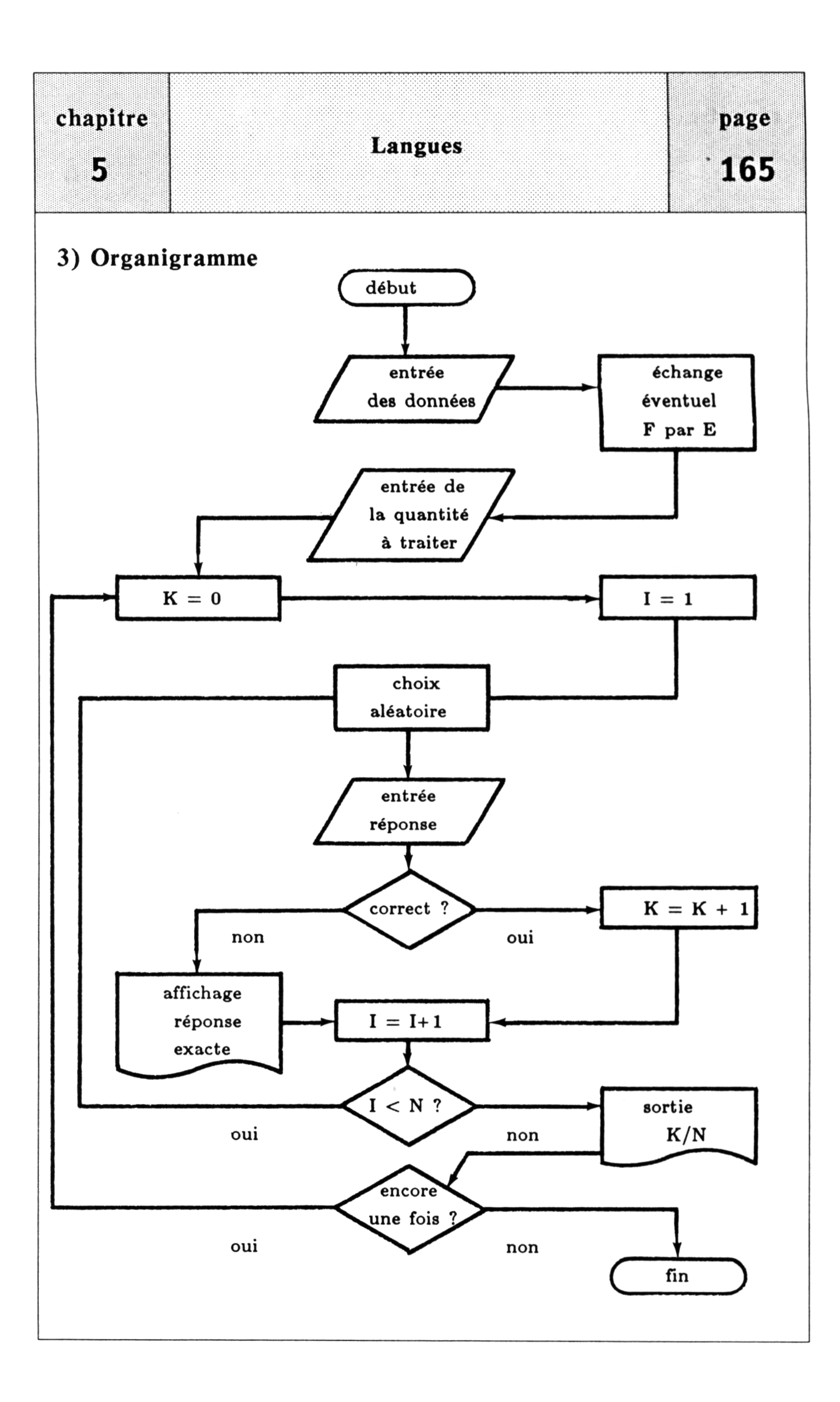
A partir du vocabulaire en stock, l'ordinateur doit faire un choix au hasard.

L'utilisateur répond à chacun des mots qui lui sont proposés. Le nombre de bonnes réponses (traductions exactes) est divisé par le nombre total de réponses à donner puis multiplié par 100, ce qui permet à l'utilisateur de connaître son pourcentage de réussite au test.

Le programme permet de recommencer le test.

Jusqu'à présent, l'analyse du problème est simple.

3) Organigramme



4) Programme

```
10 REM S3-TEST DE VOCABULAIRE
20 CLS
30 PRINT"PROGRAMME POUR LA RECITATION DE VOCABULAIRE"
40 PRINT"FRANCAIS - ANGLAIS OU VICE VERSA"
50 PRINT:PRINT:PRINT TAB(10)"PROF.W.VOSS 1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT"DANS LE STOCK DE DONNEES DE CE PROGRAMME"
80 PRINT"NE SE TROUVENT QUE DIX MOTS":PRINT
90 PRINT"SI LE PROGRAMME DOIT ETRE AGRANDI,"
100 PRINT"D'AUTRES MOTS DOIVENT ETRE AJOUTES EN DATA"
110 PRINT"DANS LES LIGNES 1000 ET SUIVANTES"
120 PRINT"LA VALEUR 'N' DOIT ETRE MODIFIE A LA LIGNE 160"
130 LOCATE 5,23:PRINT"PRESSEZ UNE TOUCHE  S.V.P"
140 A$=INKEY$:IF A$="" THEN 140
150 CLS:PRINT"UN MOMENT  S.V.P"
160 N=10
170 DIM D$(N,2)
180 FOR I=1 TO N:READ D$(I,1),D$(I,2):NEXT I
185 ZZ=1
190 CLS
200 PRINT"FRANCAIS / ANGLAIS    (1)"
210 PRINT:PRINT:PRINT
220 PRINT:PRINT:PRINT"ANGLAIS / FRANCAIS    (2)"
230 PRINT:PRINT:PRINT"ENTREZ VOTRE CHOIX  1/2":PRINT
240 INPUT"                          ";Z
245 IF Z<>ZZ THEN GOSUB 3000
250 PRINT:PRINT:PRINT"COMBIEN DE MOTS DOIVENT ETRE CONTROLES ?"
260 INPUT "NOMBRE : ";A
270 GOSUB 2000
400 PRINT:INPUT "ENCORE UNE FOIS  (O/N) ";A$
460 IF A$="O" THEN CLS:RESTORE:GOTO 180
470 PRINT:PRINT"FIN":END
1000 DATA COURIR,RUN,ECRIRE,WRITE,IMPRIMER,PRINT,ALLER,GO
1010 DATA SI,IF,SOUS-PROGRAMME,SUBROUTINE,ORDINATEUR,COMPUTER
1020 DATA CARACTERE,CHARACTER,ECRAN,SCREEN,CLE,KEY
```

```
2000 CLS
2005 K=0
2010 I=1
2020 R=INT(RND(1)*N+1)
2030 PRINT D$(R,1);:INPUT "  REPONSE : ";W$
2040 IF W$=D$(R,2) THEN K=K+1:PRINT:PRINT TAB(10) "TRES BIEN !":GOTO 2
070
2050 PRINT:PRINT:PRINT"DESOLE C'EST FAUX !!"
2060 PRINT:PRINT"LA REPONSE CORRECTE EST :";D$(R,2)
2070 PRINT:PRINT:PRINT:I=I+1
2080 IF I <= A THEN 2020
2090 KA=(K/A)*100
2100 PRINT:PRINT"POURCENTAGE DE REPONSES CORRECTES : ";KA;"%"
2110 RETURN
3000 REM ECHANGE
3010 FOR I=1 TO N
3020 H$=D$(I,1):D$(I,1)=D$(I,2):D$(I,2)=H$
3030 NEXT I
3040 RETURN
```

5) Liste des variables

A = nombre de mots à tester
A$= chaîne de caractère de réponse
D$= carnet de vocabulaire (tableau des données chaîne de caractères à 2 dimensions)
H$= idem que D$. Permet d'inverser les mots français et anglais dans le tableau pour changer le sens de traduction
I = indice de boucle
K = nombre des réponses correctes
KA= pourcentage de réponses correctes
N = nombre de mots dans le stock de données
R = nombre aléatoire
W$= réponse de l'utilisateur
Z = valeur réponse de l'utilisateur déterminant le sens de traduction
ZZ= valeur auxiliaire (initialisée à 1) pour le passage éventuel au sens de traduction anglais-français

6) Description du programme

Ligne 10-120 : titre, explications et remarques sur l'augmentation éventuelle du nombre de données du programme
Ligne 130-150 : attente, effacement de l'écran et remarque
Ligne 160-180 : indication du nombre de mots à stocker, dimensionnement et initialisation du "carnet" de vocabulaire D$, lecture des données
Ligne 185 : initialisation à 1 de la variable auxilaires ZZ
Ligne 190-240 : effacement de l'écran, affichage du menu et demande à l'utilisateur de faire son choix

Ligne 245 : test sur le choix de l'utilisateur (sens de traduction). Si le choix est différent de 1, exécution du sous programme commençant à la ligne 3000, sinon, poursuite en 250
Ligne 250-260 : demande du nombre de mots à tester
Ligne 270 : exécution du sous programme commençant à la ligne 2000
Ligne 400-420 : demande pour un nouveau test (si oui retour à la ligne 200) et fin du programme
Ligne 1000 -1020 : données
Ligne 2000-2110 : SP sélection aléatoire
Ligne 2005-2010 : initialisation des indices de boucle K et I
Ligne 2020 : nombre aléatoire
Ligne 2030 : sortie du mot et demande de la réponse traduction)
Ligne 2040 : une bonne réponse entraine un saut en 2070
Ligne 2050-2060 : message d'erreur et sortie de la réponse correcte
Ligne 2070-2080 : incrémentation de l'indice I pour le passage au mot suivant
Ligne 2090-2100 : sortie du score (pourcentage de réussite)
Ligne 2110 : retour en arrière (fin du sous programme)
Ligne 3000-3040 : SP echange : passage du français-anglais à l'anglais-français

7) Résultats

Il serait fortuit de donner ici des résultats. Ceux émis au cours du programme peuvent être immédiatement déduits de sa description (cf lignes 2040 à 2100).

5.5 Tri sur les mots

1) Présentation du problème

Le tri de données est l'un des traitements essentiels des ordinateurs. Ceci se vérifie aussi bien pour le tri des valeurs numériques que pour celui des mots ou, plus généralement, de chaînes de caractères.

Il existe toute une série d'algorithmes de tri différents. On les distingue avant tout par leur rapidité d'exécution.

La vitesse de calcul ne nous intéresse pas spécialement ici. Nous choisissons donc de présenter un algorithme qui puisse formuler clairement l'idée directrice que doit suivre toute procédure de tri.

2) Analyse du problème

Trier signifie comparer deux éléments du stock de données et les ordonner.

Dans un premier temps, on compare successivement la première chaîne de caractères en stock avec toutes les autres. Si l'ordre de classement des deux chaînes de caractères comparées n'est pas conforme à l'ordre alphabétique, ces deux chaînes sont permutées.

Après cette série de comparaisons, la première chaîne de caractères dans l'ordre alphabétique, se trouve en première position dans le stock de données.

On compare ensuite la chaîne de caractères qui se trouve en deuxième position avec toutes les autres (la première non-comprise). On procède là encore à des permutations si cela est nécessaire. A la fin de cette nouvelle série de comparaisons, la chaîne qui vient en deuxième position dans l'ordre alphabétique se trouve à la deuxième place dans l'ensemble de données.

Puis on compare la troisième chaîne de caractères avec les autres... et ainsi de suite.

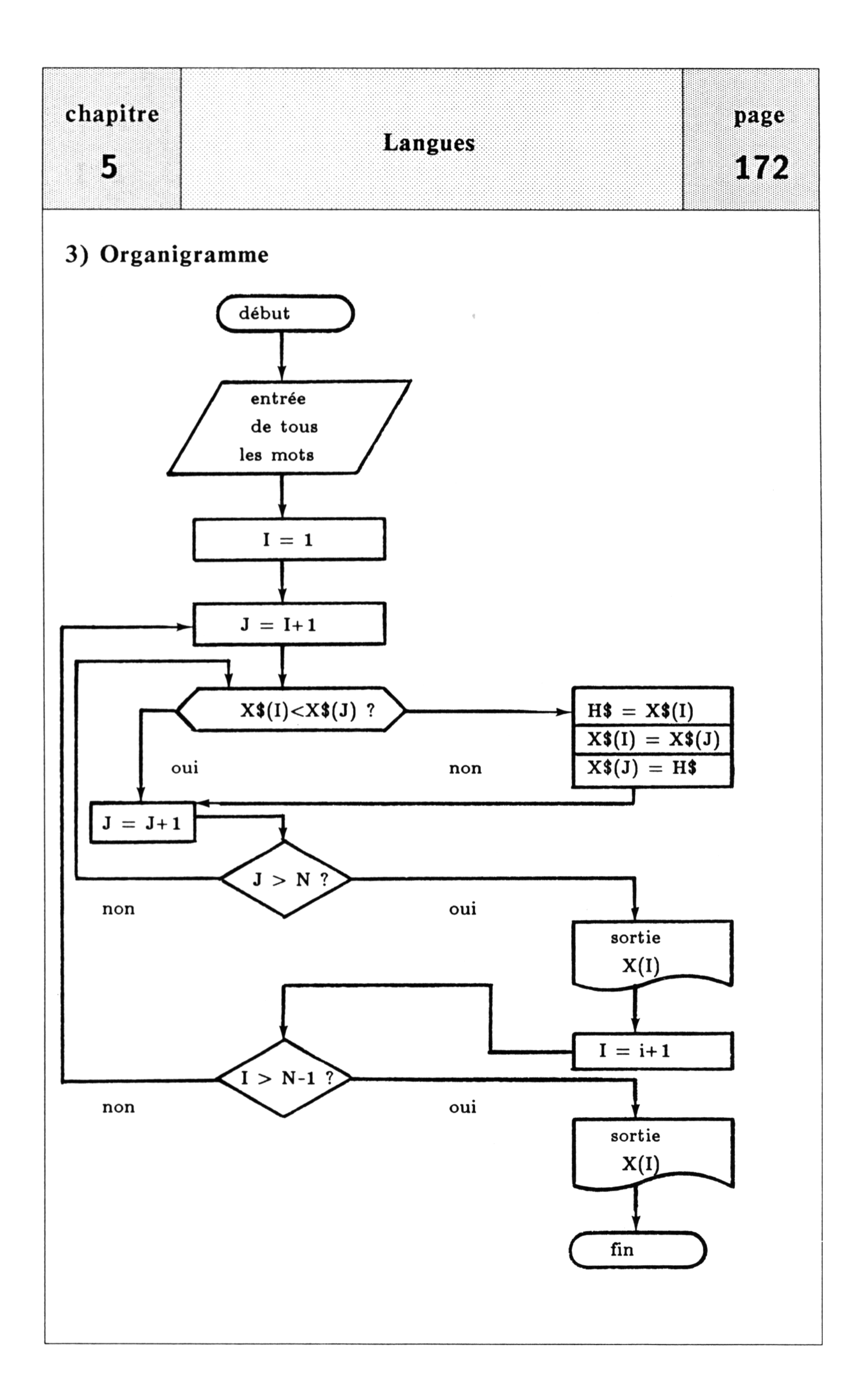
On effectue ces traitements aussi longtemps qu'il est possible de faire des comparaisons. Ainsi, si nous devons trier cinq chaînes de caractères, il nous faut en effectuer quatre. Il y a quatre comparaisons dans la première étape, trois dans la seconde, deux dans la troisième et encore une dans la quatrième.

Voici le résumé schématique de ce procédé :

1er temps :	comparaison zone 1 avec zone 2
	comparaison zone 1 avec zone 3
	comparaison zone 1 avec zone 4
	comparaison zone 1 avec zone 5
2ème temps :	comparaison zone 2 avec zone 3
	comparaison zone 2 avec zone 4
	comparaison zone 2 avec zone 5
3ème temps :	comparaison zone 3 avec zone 4
	comparaison zone 3 avec zone 5
4ème temps :	comparaison zone 4 avec zone 5

Notons que le contenu des zones est toujours susceptible de changer selon que l'on doit ou non effectuer une permutation.

3) Organigramme



4) programme

Contrairement à ce qui est présenté dans l'organigramme précédent, l'utilisateur peut décider, dans le programme suivant, si le tri des données dans l'ordre alphabétique s'applique au mots anglais ou aux mots français.

```
10 REM S4-TRI
20 CLS
30 PRINT TAB(2) "PROGRAMME DESTINE A TRIER DES MOTS"
40 PRINT:PRINT:PRINT TAB (10) "PROF.W.VOSS 1984"
50 PRINT:PRINT:PRINT:PRINT
60 PRINT"DANS CE PROGRAMME, SEULS 10 MOTS"
70 PRINT"ANGLAIS ET FRANCAIS SONT TRIES":PRINT
80 PRINT"SI D'AUTRES MOTS DOIVENT ETRE TRIES"
90 PRINT"LES RENTRER DANS LES DATA A PARTIR DE LA LIGNE 500"
100 PRINT"LA VALEUR 'N' A LA LIGNE 150 DEVRA AUSSI ETRE MODIFIEE"
110 LOCATE 5,23
120 PRINT"PRESSEZ UNE TOUCHE  S.V.P"
130 A$=INKEY$:IF A$="" THEN 130
140 CLS
150 N=10
160 DIM E$(N),D$(N)
170 FOR I=1 TO N:READ E$(I),D$(I):NEXT I
180 PRINT"FAUT IL TRIER D'APRES LES MOTS"
190 PRINT"ANGLAIS OU FRANCAIS ?"
200 PRINT:INPUT "ENTREZ  A OU F  : ";A$
210 IF A$="A" THEN 240
220 CLS:PRINT "FRANCAIS";TAB (20) "ANGLAIS":PRINT:PRINT
230 GOSUB 1000:REM TRI
235 GOTO 270
240 FOR I=1 TO N:H$=E$(I):E$(I)=D$(I):D$(I)=H$:NEXT I
250 CLS:PRINT "ANGLAIS";TAB (20) "FRANCAIS":PRINT:PRINT
260 GOSUB 1000:REM TRI
270 PRINT:PRINT"FIN":END
```

```
500 DATA IMPRIMER,PRINT,ECRIRE,WRITE,FIN,END,ALLER,GO
510 DATA COURRIR,RUN,DONNEE,DATA,SOUS-PROGRAMME,SUBROUTINE
520 DATA ENTREE,INPUT,SAUVER,SAVE,CHARGER,LOAD
1000 REM TRI
1010 FOR I=1 TO N-1
1020 FOR J=I+1 TO N
1030 IF E$(I) <= E$(J) THEN 1070
1040 H$=E$(I):G$=D$(I)
1050 E$(I)=E$(J):D$(I)=D$(J)
1060 E$(J)=H$:D$(J)=G$
1070 NEXT J
1080 PRINT E$(I);TAB(20) D$(I)
1090 NEXT I
1100 PRINT E$(I);TAB(20) D$(I)
1110 RETURN
```

5) Liste des variables

A$= variable chaîne de caractères de réponse

D$= tableau à 1 dimension des mots allemands
E$ = tableau à 1 dimension des mots anglais

G$= variable auxiliaire 1
H$= variable auxiliaire 2

I = indice de boucle
J = indice de boucle
N = nombre de mots

6) Description du programme

Ligne 10-110 : titre et explications
Ligne 120-140 : pause et effacement de l'écran
Ligne 150-170 : initialisation de la variable N. Les tableaux E et D sont dimentionnés et initialisés. Lecture des données
Ligne 180-210 : interrogation pour savoir si le tri s'effectue d'après le français ou l'anglais. Si l'on choisit l'anglais, poursuite en 270, sinon poursuite en 220
Ligne 220 : sortie de l'entête des listes à trier
Ligne 230 : saut au sous programme de tri commençant à la ligne 1000
Ligne 235 : saut à la ligne 270
Ligne 240 : permutation des mots français et anglais
Ligne 250 : sortie de l'entête des listes à trier
Ligne 260 : saut au SP de tri (1000)
Ligne 270 : fin du programme
Ligne 500-520 : données
Ligne 1000-1110 : SP tri : sous-programme destiné au tri de chaînes de caractères
1010 : indice de boucle
1020 : indice de boucle (comparaisons)
1030 : comparaison. Si elle satisfaisante, poursuite en 1070, sinon poursuite en 1040
Ligne 1040-1060 : permutation
Ligne 1070 : comparaison suivante
Ligne 1080 : une fois les comparaisons effectuées, sortie des deux chaînes de caractères qui se trouvent maintenant en Ième position dans l'ensemble de données
Ligne 1090 : traitement suivant

Ligne 1100 : sortie de la dernière paire de chaînes de caractères
Ligne 1110 : retour en arrière (fin du sous programme)

7) Résultats

Si l'on souhaite que le tri se fasse sur les mots français, nous obtenons :

FRANCAIS	ANGLAIS
ALLER	GO
DONNEE	DATA
FIN	END
IMPRIMER	PRINT
RECEVOIR	GET
...	...

BIOLOGIE
ECOLOGIE



6.1 Avant-Propos

En biologie et en écologie, la fonction principale de l'ordinateur est de résoudre les problèmes que pose la simulation des processus de développement et de croissance. Il est, bien sûr, possible d'élaborer des programmes d'interrogation d'après le modèle donné dans le chapitre précédent. Nous avons également la possibilité de sortir des données à l'écran sous forme de tableaux ou de graphiques.

6.2 Croissance continue

1) Présentation du problème

La population d'un pays ayant un taux de croissance initial de X0 possède un taux annuel de croissance de R % (R est positif). Si cette croissance est régulière, ce qui à long terme est naturellement très improbable, elle devient exponentielle.

Pour écrire un programme en BASIC nous allons simuler une procédure de croissance, très simple, semblable à celle que nous venons d'exposer.

2) Analyse du problème

Il est nécessaire d'entrer le taux initial et le taux annuel de croissance.

Au bout d'un an, on obtient le taux X1 suivant :

X1 = X0 + X0 * R/100 = X0 * (1 + R/100)

Deux ans après, X2 :

X2 = X1 * (1 + R/100)= X0 * (1 + R/100) * (1 + R/100)

Il s'ensuit que :

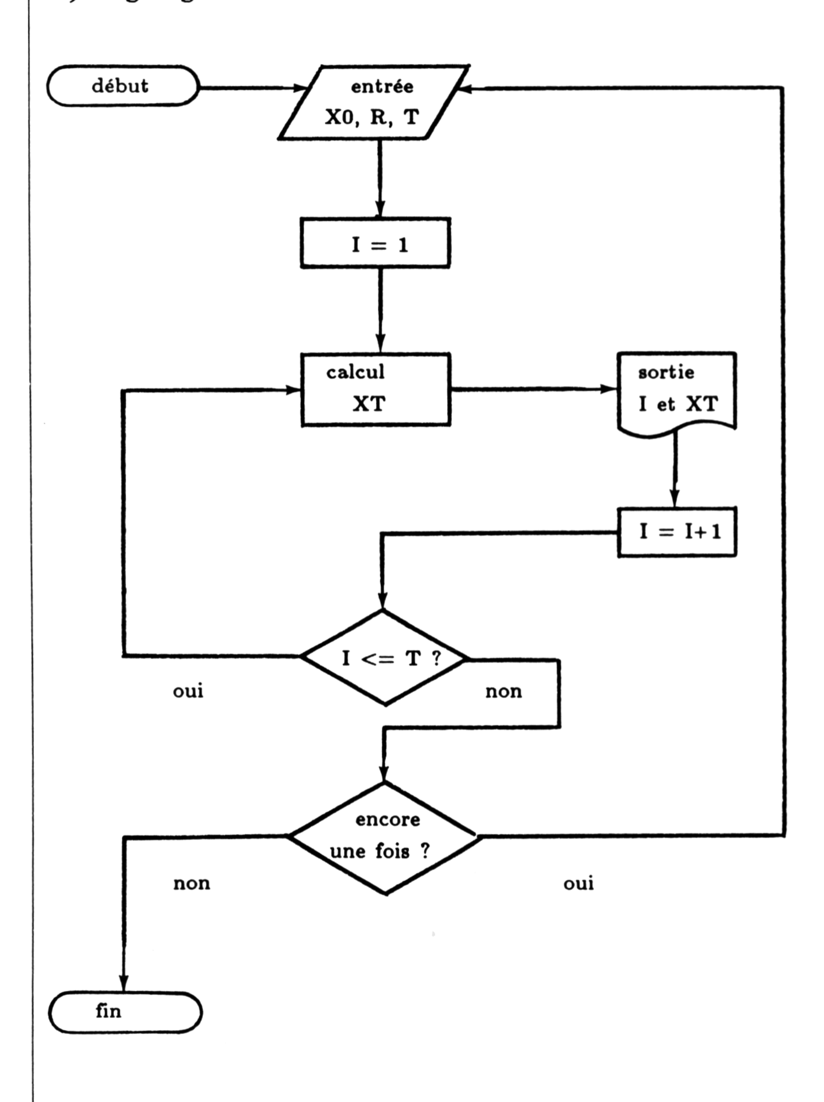
$$X2 = X0 * (1 + R/100)^2$$

Pour T années la formule est donc :

$$XT = X0 * (1 + R/100)^T$$

Cette équation correspond à l'algorithme de calcul employé dans le programme BASIC correspondant.

3) Organigramme



4) Programme

```
10 REM B1-CROISSANCE
20 CLS
30 PRINT"PROGRAMME POUR LA REPRESENTATION PAR DES"
40 PRINT"TABLEAUX  DE LA CROISSANCE NON FREINEE"
50 PRINT:PRINT:PRINT TAB(10) "PROF.W.VOSS 1984"
60 PRINT:PRINT:PRINT:PRINT
70 INPUT "TAUX INITIAL                    : ";XO
80 PRINT:INPUT "TAUX DE CROISSANCE            : ";R
90 PRINT:INPUT "NOMBRE D'ANNEE                : ";T
100 CLS:PRINT"APRES... ANNEES    TAUX":PRINT:PRINT
110 I=1
120 XT=XO*(1+R/100)^I
130 PRINT TAB(5)I;TAB(19) USING "#####.##";XT
140 I=I+1
145 IF I/10 = INT (I/10) THEN PRINT"ENTREZ LA COMMANDE <CONT>  S.V.P":
STOP
150 IF I <= T THEN 120
160 PRINT:PRINT:INPUT "UN AUTRE CALCUL  (O/N) ";A$
170 IF A$="O" THEN CLS:GOTO 70
180 PRINT:PRINT:PRINT"FIN":END
```

5) Liste des variables

A$= chaîne de caractères de réponse (oui, non)
I = indice de boucle
R = taux d'accroissement en %
T = nombre d'années
X0= taux initial
XT= taux au moment T

6) Description du programme

Ligne 10-60 : titre
Ligne 70-90 : demande des informations dont le programme a besoin
Ligne 100 : sortie de l'entête du tableau
Ligne 110 : initialisation de l'indice I de boucle à 1 (1ère année)
Ligne 120 : calcul du taux
Ligne 130 : sortie
Ligne 140 : incrémentation de 1 de l'indice I (passage à l'année suivante)
Ligne 145 : le programme est interrompu tous les dix ans (lorsque le reste de la division de I par 10 est nul) nous ne pouvons poursuivre qu'avec la commande CONT (message)
Ligne 150 : le programme se déroule tant que I est inférieur ou égal à T (ligne 120)
Ligne 160-180 : fin du programme à moins qu'une exécution supplémentaire ne soit souhaitée (dans ce cas retour à la ligne 70 après effacement de l'écran)

7) Résultats

Entrons par exemple un taux initial de valeur 500, un taux d'accroissement de 4 (%) et une durée de 8 (ans par exemple), le programme affiche :

APRES ..... ANNEES	TAUX
1	520.00
2	540.80
3	562.43
4	584.93
5	608.33
6	632.66
7	657.97
8	684.28

ENCORE UNE FOIS (0/N) ?

6.3 Baisse de croissance

1) Présentation du problème

Telle qu'elle a été présentée dans la section précédente, la croissance exponentielle n'existe pratiquement pas dans la réalité ou seulement pour des périodes très limitées. On rencontre plus souvent des fonctions stables qui croissent, en général, avec l'augmentation de la valeur des variables concernées.

Référons nous, par exemple, à une population dont la croissance est exponentielle. Elle se heurte tôt ou tard à un autre facteur : celui de sa capacité de production de biens de première nécessité. En admettant que la démographie ne cesse de croître, la croissance se transforme alors en dépression puis en crise à plus ou moins brève échéance.

Le programme BASIC suivant va simuler cette dépression.

2) Analyse du problème

Notons que l'analyse qui suit rejoint celle de la section précédente :

Nous utilisons ici aussi des valeurs en entrée :

X0 = population intiale
R0 = taux de croissance de la première année (en %)

A l'inverse de l'exemple précédent, nous partons du fait que le taux de croisssance annuelle (en baisse) croît de façon inverse mais proportionnellement à la population (en hausse).

Etant donné que la croissance de la population est progressive dans le temps, il en est de même pour la baisse du taux de croissance.

Ainsi, pour une période deux fois plus importante, le taux de croissance est divisé par deux, pour une période quatre fois plus importante, le taux est divisé par quatre, etc.

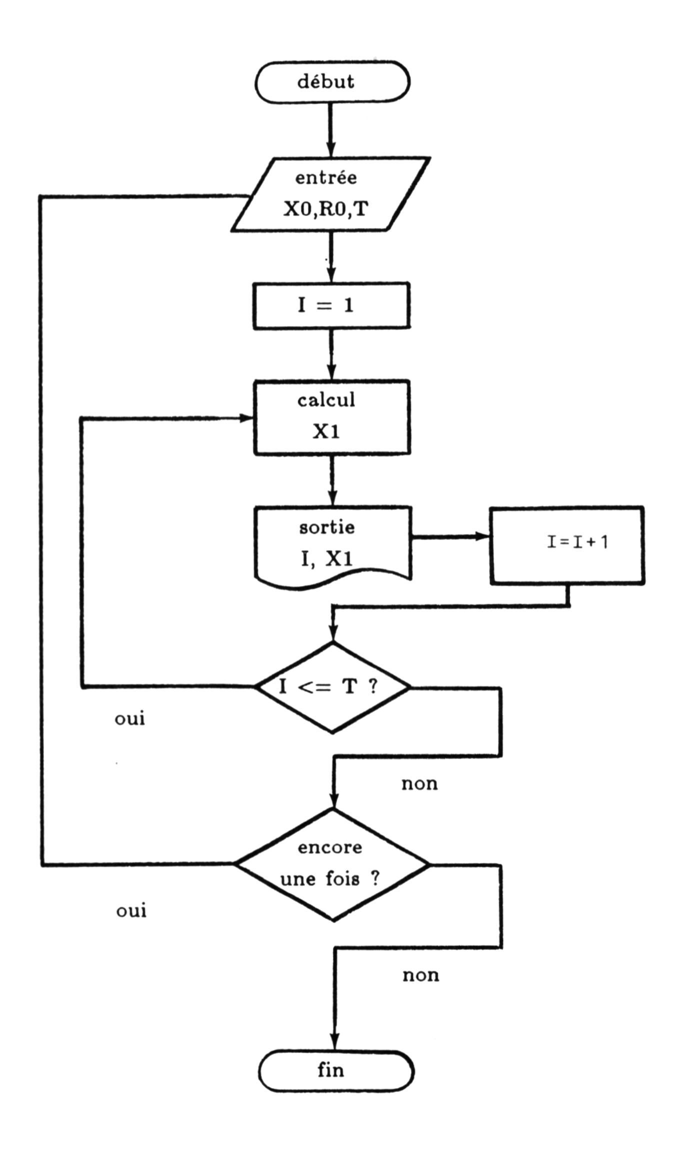
Il n'est pas utile ici de présenter d'autes méthodes simulant cette dépression.

L'algorithme repose sur la formule mathématique suivante :

X1 = X0 * (1 + R/I)

Ce qui signifie qu'une population X1 à une période donnée, est fonction de la population X0 de l'année précédente comme nous l'avons vu dans la dernière section. De plus, le taux de croissance est divisé par l'indice de boucle de manière à ce que ce taux devienne d'autant plus petit que I augmente. Ceci marque l'avancement du temps.

3) Organigramme



4) Programme

```
10 REM B2-CROISSANCE FREINEE
20 CLS
30 PRINT"PROGRAMME POUR LA REPRESENTATION PAR UN "
40 PRINT TAB(2)"TABLEAU D'UNE PROCEDURE DE CROISSANCE FREINEE "
50 PRINT:PRINT:PRINT TAB(9)"PROF.  W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT:PRINT
70 INPUT"TAUX INITIAL      : ";X0
80 PRINT:INPUT"TAUX DE CROISSANCE A LA PREMIERE ANNEE (%)  : ";R
85 PRINT:INPUT"COMBIEN D'ANNEES  : ";T
90 CLS
100 PRINT"APRES ... ANNEES    TAUX    CROISSANCE (%)":PRINT:PRINT
110 I=1
120 X1=X0+X0*(R/I)/100
140 Z=(X1-X0)/X0:Z=Z*100
150 PRINT TAB(5)I;TAB(19) USING "####.##";X1;:PRINT TAB(28) USING "###
#.##";Z
160 X0=X1:I=I+1:IF I<=T THEN 120
180 PRINT:INPUT"ENCORE UNE FOIS ? (O/N)";A$
190 IF A$="O" THEN CLS:GOTO 70
200 PRINT:PRINT"FIN DE LA SORTIE":END
```

5) Liste des variables

A$= chaîne de caractères de réponse (Oui/Non)
I = indice de boucle (nombre d'années)
R = taux de croissance initial de la première année
T = durée de l'étude
X0= population à une période donnée
X1= même population l'année suivante
Z = taux de croissance de l'année en cours par rapport à l 'année précédente

6) Description du programme

Ligne 10-60 : titre et explications
Ligne 70-90 : demande des données d'entrée et effacement de l'écran
Ligne 100 : affichage de l'entête d'un tableau
Ligne 110 : initialisation à 1 de l'indice de boucle I
Ligne 120-150 : calcul de la population du moment, et de son taux de croissance par rapport à l'année précédente. Sortie de ces résultats (au centième près) et de la période à laquelle ils correspondent.
Ligne 160 : X0 prend la valeur de X1. Poursuite des calculs tant que I, après incrémentation de 1, reste inférieur ou égal à la durée de l'étude désirée
Ligne 180-200 : interrogation pour savoir si un autre calcul est souhaité (si oui, retour en 70) et fin du programme

7) Résultats

Entrons par exemple comme taux initial la valeur 500 et comme taux de croissance économique de la première année la valeur 4 (%), indiquons enfin que l'étude doit durer 8 ans. Nous obtenons :

APRES ... ANNEES	TAUX	CROISSANCE(%)
1	520.00	4.00
2	530.40	2.00
3	537.47	1.33
4	542.84	1.00
5	547.18	0.80
6	550.83	0.67
7	553.98	0.57
8	556.75	0.50

ENCORE UNE FOIS (O/N) ?

6.4 Pollution de l'environnement

1) Présentation du problème

En règle générale les populations qui ne cessent de croître entravent elles-mêmes leur développement en dégradant de plus en plus leurs conditions de vie.

Prenons l'exemple de la pollution de l'environnement au cours des phases de l'industrialisation.

Ici aussi apparaît un "frein à la croissance" comparable à celui développé dans la section précédente.
Mais contrairement à l'analyse de la section 6.3, ce "frein" ne dépend pas directement de la population mais d'une variable qui lui est proportionnelle.

2) Analyse du problème

La population de la période 1 ici n'est plus seulement fonction de celle de la période 0 et de son taux d'accroissement. Ainsi, à la place de l'équation :

X1 = X0 + X0 * R/100

nous avons :

X1 = X0 + X0 * (R/100 - P * B0)

Où B0 représente le degré de pollution de l'environnement au moment 0. A la valeur B0 est associé un paramètre P approprié (par ex. P = 0.01) qui contribue à la diminution de la croissance.

D'après ce qui précède, nous pouvons écrire :

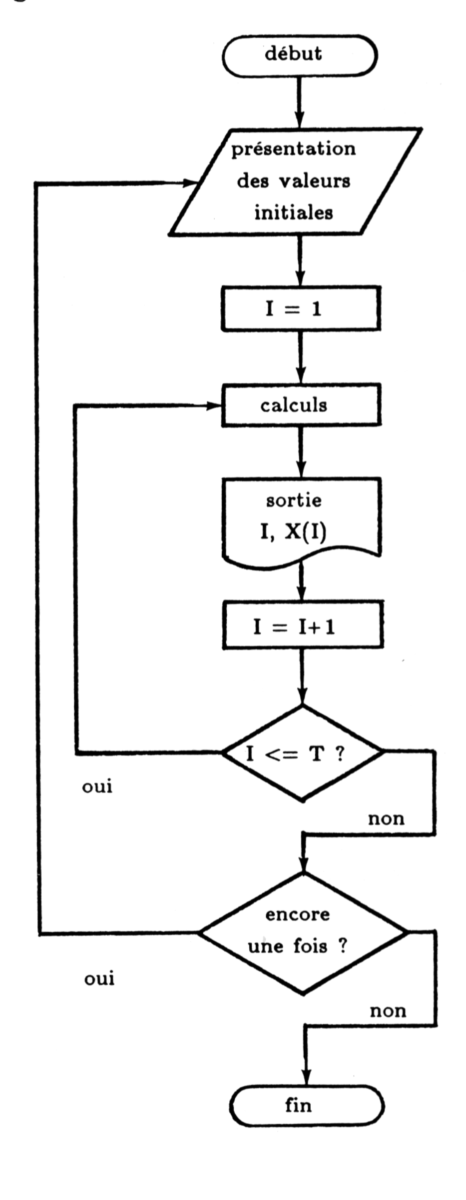
B1 = B0 + A * X0

Si nous prenons B0 = 0 comme degré de pollution initial, la pollution augmente alors en fonction de la taille de la population dont il nous faut fixer le coefficient associé (par ex. A = 0.015).

On a donc les relations :

(1) B1 = B0 + A * X0
(2) X1 = X0 + X0 * (R/100-P*B1)

3) Organigramme



4) Programme

```
10 REM B3-ENVIRONNEMENT
20 CLS
30 PRINT"PROGRAMME POUR LA REPRESENTATION PAR"
40 PRINT"UN TABLEAU D'UNE PROCEDURE DE CROISSANCE"
50 PRINT"QUI EST FREINEE PAR SA PROPRE DYNAMIQUE"
60 PRINT:PRINT:PRINT:PRINT TAB (10)"PROF.W.VOSS 1984"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"CE PROGRAMME REQUIERT LES INFORMATIONS"
90 PRINT"SUIVANTES   : ":PRINT
100 PRINT:INPUT "TAUX DE SORTIE                    : ";X0
105 PRINT:INPUT "COMBIEN D'ANNEE                   : ";T
110 PRINT:INPUT "TAUX DE CROISSANCE ANNUEL EN %  : ";R
120 PRINT:INPUT "FACTEUR DE POPULATION             : ";P
130 CLS
140 PRINT"TEMPS      TAUX      CROISSANCE (%)     POPULATION":PRINT:P
RINT
150 I=1:A=0.015:B0=0
160 B1=B0+A*X0
170 X1=X0+X0*(R/100-P*B0)
175 IF X1 <= 0 THEN PRINT"LA POPULATION   EST   DETRUITE":GOTO 210
180 Z=(X1-X0)/X0:Z=Z*100
190 PRINT TAB(2) I; TAB(11) USING "#####.##";X1;:PRINT TAB(25) USING "
####.##";Z;
195 PRINT TAB(42) USING "#####.##";B1
200 I=I+1:IF I <= T THEN B0=B1:X0=X1:GOTO 160
210 PRINT:PRINT:INPUT "ENCORE UN CALCUL  (O/N) ";A$
220 IF A$ = "O" THEN CLS:GOTO 80
230 PRINT:PRINT"FIN":END
```

5) Liste des variables

A = facteur degré de pollution conditionnée par la population
A$= variable chaîne de caractères de réponse (Oui, Non)
B0 = degré de pollution initial
B1 = degré de pollution à l'année suivante
I = indice de boucle
P = facteur degré de pollution
R = taux de croissance annuel
T = durée de l'étude
X0= population à une période donnée
X1= même population l'année suivante
Z = taux de croissance de l'année en cours par rapport à l'année précédente

6) Description du programme

Ligne 10-70 : titre et explications
Ligne 80-120 : demande des informations d'entrée
Ligne 130-140 : effacement de l'écran et sortie de l'entête du tableau
Ligne 150 : initialisation à 1 de l'indice de boucle I, à 0 du degré de pollution initial B0, à 0.015 du facteur de degré de pollution A
Ligne 160-170 : calculs pour l'année suivante
Ligne 175 : si le taux de croissance de la population calculée est négatif ou nul, un message apparait à l'écran puis saut à la ligne 210
Ligne 180 : calcul de 2
Ligne 190-195 : sortie des résultats de l'année en cours
Ligne 200 : incrémentation de l'indice de boucle I, tant que I reste inférieur à T, X0 prend la valeur de X1, B0 celle de B1 et retour à ligne 160

Ligne 210-230 : interrogation pour savoir si un autre calcul est souhaité (si oui, effacement de l'écran, retour à la ligne 80) et fin du programme

7) Résultats

Entrons par exemple un taux de croissance initial de 1000, une durée de 8 ans, un taux de croissance annuelle de 5 % et un facteur de degré de pollution de 0.01. On obtient :

TEMPS	TAUX	CROISSANCE %	POLLUTION
1	1050.00	5.00	15.00
2	945.00	-10.00	30.75
3	701.66	-25.75	44.93
4	421.52	-39.93	55.45
5	208.87	-50.45	61.77
6	90.29	-56.77	64.91
7	36.20	-59.90	66.26
8	14.02	-61.26	66.80

ENCORE UNE FOIS (O/N) ?

HISTOIRE
GEOGRAPHIE

7

7.1 Avant-Propos

Nous allons établir ici, comme dans le chapitre sur les Langues, des programmes de test et d'interrogation. Ici aussi, pour les exemples, nous nous limitons dans le nombre des données, puisque c'est la construction des programmes et leur fonctionnement qui nous intéressent.

Nous verrons comment établir et entrer des fichiers de données plus importants, et adapter selon les commandes DATA le nombre N de ces données en vue de l'utilisation de ces programmes dans la pratique.

7.2 Dates historiques

1) Présentation du problème

Le programme qui est ici présenté permet à l'élève d'apprendre plus facilement les dates historiques et de contrôler ses connaissances.

En pratique, l'ordinateur présente des évènements historiques à l'utilisateur et lui demande d'indiquer les dates auxquelles ils se sont déroulés. L'utilisateur décide combien de questions va lui poser l'ordinateur par série de tests.

Le programme est conçu de telle manière qu'il est capable de dire si la reponse est juste ou, dans le cas contraire, de redonner une chance à l'élève. Si l'utilisateur préfère abandonner, la réponse correcte apparait à l'écran.

2) Analyse du problème

Nous cernons suffisamment le problème pour savoir comment procéder dans ce programme.

Nous avons besoin en entrée d'une information permettant à l'utilisateur de dire combien de questions il veut qu'on lui pose. Le programme doit à chaque fois contrôler les réponses données, ce qui ne pose aucune difficulté particulière.

3) Organigramme

début

entrée
des données

entrée
Z = nombre
de tests

I = 1

sélection
aléatoire

entrée date

exact ?

non

oui

message

encore
une fois ?

oui

non

I = I+1

affichage
chiffre
correct

I <= Z ?

non

encore ?

non

oui

fin

4) programme

```
10 REM E1-DATES HISTORIQUES
20 CLS
30 PRINT"PROGRAMME DE RECHERCHE DES DATES HISTORIQUES"
40 PRINT:PRINT:PRINT TAB (10) "PROF.W.VOSS 1984"
50 PRINT:PRINT:PRINT:PRINT
60 PRINT"CE PROGRAMME PROPOSE UN CHOIX"
70 PRINT"DE 10 DATES HISTORIQUES"
80 PRINT"PLACEES EN 'DATA' A PARTIR DE"
90 PRINT"LA LIGNE 500"
100 PRINT"CETTE LISTE DE DATES PEUT ETRE ALLONGEE"
110 PRINT"ET LA VALEUR 'N', LIGNE 150, DOIT ETRE MODIFIEE"
120 LOCATE 5,23
130 PRINT"PRESSEZ UNE TOUCHE  S.V.P"
140 A$=INKEY$:IF A$="" THEN 140
150 N=10
155 DIM J$(N),S$(N)
160 CLS
170 FOR I=1 TO N: READ J$(I),S$(I):NEXT I
180 PRINT"COMBIEN DE QUESTIONS VOULEZ-VOUS ?"
190 PRINT:INPUT "INDIQUEZ LE NOMBRE S.V.P  : ";Z
200 FOR I=1 TO Z
202 R=INT (RND(1)*N+1)
205 CLS
210 PRINT"EN QUELLE ANNEE: ":PRINT
230 PRINT S$(R):PRINT
240 PRINT:PRINT:INPUT "ANNEE";K$
250 IF K$=J$(R) THEN PRINT:PRINT TAB(9) "B R A V O":PRINT:GOTO 300
260 PRINT:PRINT"CE N'EST PAS LA BONNE REPONSE !":PRINT
270 PRINT:PRINT"ENCORE UN ESSAI  (O/N) ";:INPUT A$
280 IF A$="O" THEN 205
290 PRINT:PRINT"LA DATE EXACTE EST :";J$(R)
300 FOR II=1 TO 2000:NEXT II:NEXT I
310 PRINT:PRINT:PRINT"UN AUTRE TEST  (O/N) ";:INPUT A$
320 IF A$="O" THEN CLS: GOTO 180
330 PRINT:PRINT"FIN":END
```

```
500 DATA 1789,REVOLUTION FRANCAISE
505 DATA 800,CHARLEMAGNE SACRE EMPEREUR
510 DATA 1776,INDEPPENDANCE DES ETATS UNIS
515 DATA -44,FIN DU REGNE DE JULES CESAR
520 DATA 732,CHARLES MARTEL ARRETE L'INVASION ARABE
525 DATA 1963,ASSASSINAT DE J.F.KENNEDY
530 DATA 1917,REVOLUTION D'OCTOBRE EN RUSSIE
535 DATA 1969,PREMIER HOMME SUR LA LUNE
540 DATA 1815,WATERLOO
545 DATA 1515,MARIGNAN
```

5) liste des variables

A$= variable chaîne de caractères de réponse
I = indice de boucle
J$ = dates
N = nombre de données stockées
R = nombre aléatoire
S$ = énoncés des faits historiques
Z = nombre de questions à poser par test

6) Description du programme

Ligne 10-120 : titre et explications
Ligne 130-140 : attente
Ligne 150-170 : initialisation du nombre de données à stocker, dimensionnement des tableaux dans lesquels seront stockées les données, effacement de l'écran et initialisation des tableaux par lecture des données
Ligne 180-190 : demande du nombre de questions souhaitées
Ligne 200-300 : tests
Ligne 202-240 : choix aléatoire d'un évènement historique par l'ordinateur et demande d'une date (réponse) à l'utilisateur
Ligne 250 : si la réponse est correcte message et poursuite en 300
Ligne 260-280 : si la réponse n'est pas correcte, message et demande si l'utilisateur désire avoir une deuxième chance (si oui, retour en 205)
Ligne 290 : si non, sortie de la bonne réponse
Ligne 300 : boucle d'attente puis retour à la ligne 200
Ligne 310-330 : demande si un deuxième déroulement du test est souhaité, si oui, retour à la ligne 180 après effacement de l'écran, si non, fin du programme
Ligne 500-590 : données

7) Résultats

Il n'est pas indispensable ici de présenter les résultats. Il est plus utile que le lecteur teste le programme en fonction des données présentes et qu'il étende ensuite ce programme comme il le desire.

7.3 Départements et préfectures

1) Présentation du problème

Le programme présenté dans cette section est également un programme d'entrainement. Il s'agit, là encore, pour l'utilisateur de répondre aux questions posées par l'ordinateur. Le programme soumet, au hasard, le nom d'un département français à l'utilisateur qui doit alors entrer le nom de sa préfecture. Si au contraire l'ordinateur présente des préfectures, l'utilisateur doit donner les départements correspondants.

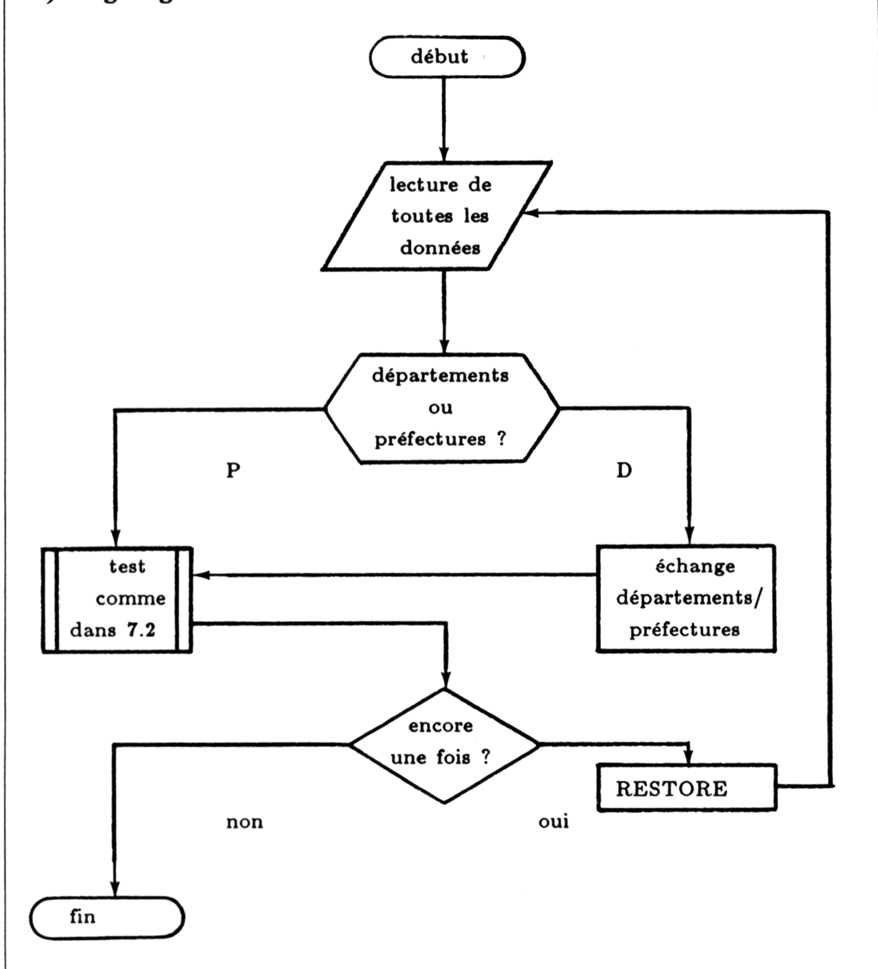
Notons que ce programme ne présente pas de particularités par rapport au précédent. Nous nous limitons donc à une explication brève du problème. Ce programme doit, lui aussi, être considéré à titre d'exemple.

L'utilisateur peut, sans difficulté, intégrer d'autres données au programme. Il peut ainsi constituer un champs d'applications suffisamment large pour étudier plusieurs matières scolaires.

2) Analyse du problème

Etant donné qu'aucun problème nouveau ne surgit ici, nous pouvons directement tracer l'organigramme.

3) Organigramme



4) Programme

```
10 REM E2-PREFECTURES ET DEPARTEMENTS
20 CLS
30 PRINT"PROGRAMME POUR LA VERIFICATION DES CONNAISSANCES"
40 PRINT"EN GEOGRAPHIE"
50 PRINT:PRINT:PRINT TAB(10) "PROF.W.VOSS 1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT"CE PROGRAMME RECLAME APRES INDICATION"
80 PRINT"LA PREFECTURE POUR UN DEPARTEMENT OU"
90 PRINT"LE DEPARTEMENT POUR UNE PREFECTURE":PRINT:PRINT
100 PRINT"AVEC D'AUTRES DONNEES DANS LES"
110 PRINT"LIGNES DE DATA 500 ET SUIVANTES ET AVEC LA"
120 PRINT "LA MODIFICATION DE 'N' A LA LIGNE 160, LE PROGRAMME"
130 PRINT"PEUT ETRE UTILISE A D'AUTRES FIN"
140 LOCATE 5,23:PRINT"PRESSEZ UNE TOUCHE  S.V.P  "
150 A$=INKEY$:IF A$="" THEN 150
160 N=11
170 DIM L$(N),S$(N)
180 FOR I=1 TO N:READ L$(I),S$(I):NEXT I
190 CLS
200 PRINT"FAUT IL PRESENTER LES PREFECTURES OU"
210 INPUT "DEPARTEMENTS  (P/D) ";A$
215 C$="PREFECTURES"
220 IF A$="D" THEN C$="DEPARTEMENT":GOTO 250
230 GOSUB 1000:REM TEST
240 GOTO 270
250 FOR I=1 TO N:H$=L$(I):L$(I)=S$(I):S$(I)=H$:NEXT I
260 GOSUB 1000:REM TEST
270 PRINT:PRINT:INPUT "ENCORE UNE FOIS  (O/N)  ";A$
280 IF A$="O" THEN CLS:RESTORE:GOTO 180
290 PRINT:PRINT"FIN":END
500 DATA ALLIER,MOULINS,CREUSE,GUERET,GERS,AUCH
510 DATA LOIRE,SAINT ETIENNE,LOT,CAHORS
520 DATA MANCHE,SAINT LO,NORD,LILLE
530 DATA RHONE,LYON,SARTHE,LE MANS,SAVOIE,CHAMBERY
540 DATA SOMME,AMIENS
```

```
1000 REM TEST
1010 CLS
1020 PRINT:INPUT "COMBIEN DE QUESTIONS : ";Z
1030 FOR I=1 TO Z
1040 R=INT(RND(1)*N+1)
1050  PRINT:PRINT L$(R):PRINT:PRINT:PRINT C$;" : ";:PRINT:PRINT:INPUT
X$
1060 IF X$=S$(R) THEN PRINT:PRINT TAB (10) "CORRECT   ":PRINT:GOTO 110
0
1070 PRINT:PRINT TAB (10) "ERREUR !":PRINT
1080 PRINT:PRINT"LA BONNE REPONSE EST  :"
1090 PRINT:PRINT TAB (10) S$(R):PRINT
1100 NEXT I
1110 RETURN
```

5) Liste des variables

A$= variable chaîne de caractères de réponse
C$= variable auxiliaire qui, selon la sélection, correspond à la préfécture ou au département
I = indice de boucle
L$= départements
N = nombre de données
S$= préfectures

6) Description du programme

Ligne 10-130 : titre et explications
Ligne 140-150 : pause
Ligne 160-180 : lecture de l'ensemble des données
Ligne 190-210 : question pour orienter le test sur les préfectures ou sur les départements
Ligne 220 : si le test porte sur les préfectures, la variable auxiliaire C$ correspond aux départements et on passe à la ligne 250
Ligne 230 : saut dans le sous-programme 1000, dans lequel le test s'effectue
Ligne 240 : saut à la ligne 270
Ligne 250 : permutation des départements et des préfectures afin d'utiliser le même sous programme quelque soit la sélection. (190-210)
Ligne 260 : voir ligne 230
Ligne 270-290 : interrogation pour savoir si une autre série de tests est souhaitée. Si oui, effacement de l'écran et retour à la ligne 180 une fois que le stock de données a été restauré (ceci est nécessaire pour annuler l'échange éventuel de la ligne 250 et pour recréer les conditions initiales). Sinon fin du programme
Ligne 500-540 : données

Ligne
1000-1110 : sous-programme test :
questionnaire selon le nombre de questions souhaitées. Message de bonne réponse et d'erreur. Les détails correspondent (en principe) à ceux de la section 7.2. Nous n'en dirons donc pas davantage sur ce sous-programme

7) Résultats

Là non plus, il n'est pas nécessaire de donner les résultats de ce programme. L'examiner ou mieux encore le tester montre d'une manière très claire ce qui se passe dans le détail.

7.4 L'évolution démographique de différents pays

1) Description du problème

Le programme présenté dans cette section établit des prévisions sur l'évolution démographique de différents pays jusqu'en l'an 2000.

Nous avons sélectionné parmi les pays ou ensembles de pays :

monde (comprenant tous les pays)

pays industrialisés
pays en voie de développement

République Populaire de Chine
Inde
Japon
USA
URSS
République Fédérale d'Allemagne
RDA

2) Analyse du problème

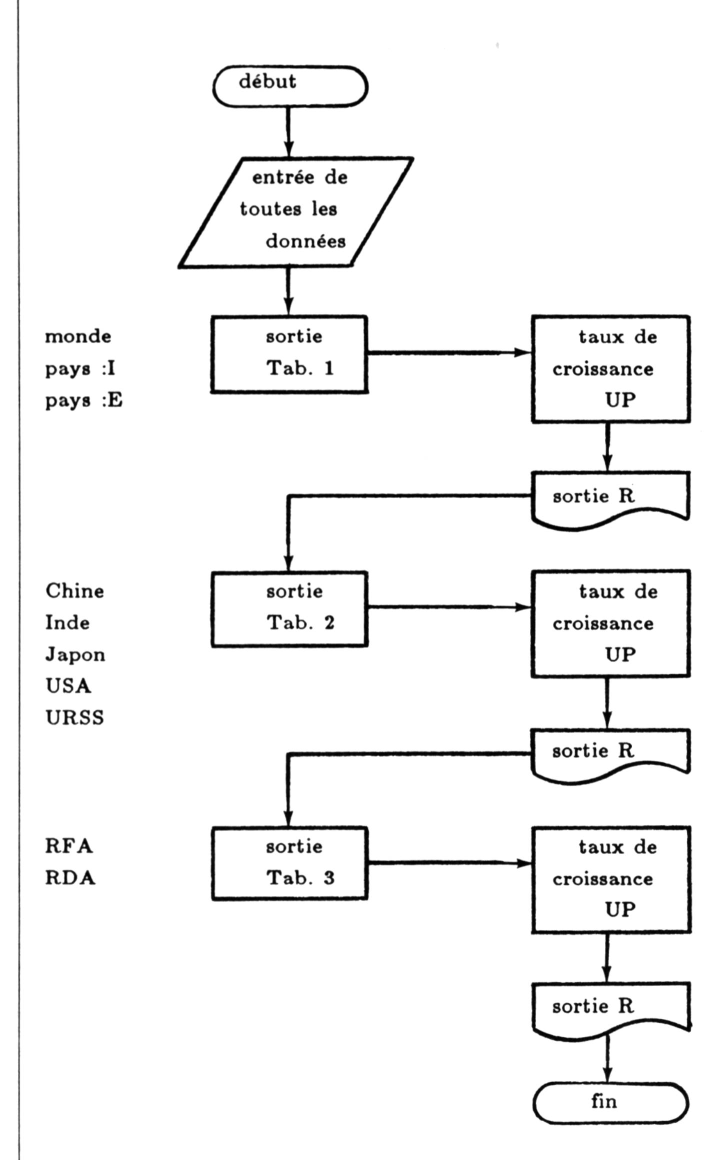
Pour l'étude des prévisions démographiques, il est nécessaire de suivre une méthode précise.

De nombreuses méthodes ont été publiées mais nous ne prenons ici qu'un exemple, celui du rapport GLOBAL 2000 (Washington 1980).

A partir de ces données, nous calculons également le taux de croissance moyen par an pour disposer d'une information supplémentaire.

Aucune difficulté particulière n'apparaît en ce qui concerne la logique du programme, cependant, plusieurs interruptions du programme sont nécessaires du fait de l'abondance des résultats.

3) Organigramme



4) Programme

```
10 REM E4-POPULATION
20 CLS
30 PRINT"PROGRAMME POUR LA REPRESENTATION PAR UN TABLEAU"
40 PRINT"DE L'EVOLUTION DEMOGRAPHIQUE DE "
50 PRINT TAB(12)"DIFFERENT PAYS"
60 PRINT:PRINT:PRINT TAB(9)"PROF.  W.VOSS, 1984"
70 DIM B(10,6),J(6),L$(10),WR(10)
80 FOR I=1 TO 6:READ J(I):NEXT I
90 FOR L=1 TO 10:FOR I=1 TO 6:READ B(L,I):NEXT I:NEXT L
100 FOR L=1 TO 10:READ L$(L):NEXT L
110 GOSUB 1000: REM ATTENDRE
140 PRINT"ANNEE   MONDE    PAYS I        PAYS E"
150 GOSUB 1200: REM TRAIT
160 PRINT
170 FOR I=1 TO 6
180 PRINT J(I);TAB(8)B(1,I);TAB(18)B(2,I);TAB(32)B(3,I):PRINT
190 NEXT
195 GOSUB 1200: REM TRAIT
200 FOR L=1 TO 3
210 X=B(L,1):Y=B(L,6)
220 GOSUB 1100: REM TAUX DE CROISSANCE
230 NEXT L
240 PRINT:PRINT"TAUX %";:PRINT TAB(8)WR(1);TAB(18)WR(2);TAB(32)WR(3):P
RINT
250 PRINT:GOSUB 1000: REM ATTENDRE
260 CLS
270 PRINT"ANNEE  CHINE INDE   JAPON   USA   URSS"
280 GOSUB 1200: REM ATTENDRE
290 FOR I=1 TO 6
300 PRINT J(I);TAB(8)B(4,I);TAB(14)B(5,I);TAB(21)B(6,I);
302 PRINT TAB(28)B(7,I);TAB(34)B(8,I):PRINT
310 NEXT I
320 GOSUB 1200: REM TRAIT
330 FOR L=4 TO 8
340 X=B(L,1):Y=B(L,6)
```

```
350 GOSUB 1100: REM TAUX DE CROISSANCE
360 NEXT L
370 PRINT:PRINT"%";
380 PRINT TAB(8)WR(4);TAB(14)WR(5);TAB(21)WR(6);TAB(28)WR(7);TAB(34)WR
(8)
385 PRINT:PRINT
390 GOSUB 1000: REM ATTENDRE
400 CLS
410 PRINT"ANNNEE      RFA          RDA ":PRINT
420 GOSUB 1200: REM TRAIT
430 FOR I=1 TO 6
440 PRINT J(I);TAB(12)B(9,I);TAB(24)B(10,I):PRINT
450 NEXT I
460 GOSUB 1200: REM TRAIT
470 FOR L=9 TO 10
480 X=B(L,1):Y=B(L,6)
490 GOSUB 1100: REM TAUX DE CROISSANCE
492 NEXT L
494 PRINT"%";:PRINT TAB(12)WR(9);TAB(24)WR(10)
496 PRINT:PRINT"FIN":END
500 DATA 1975,1980,1985,1990,1995,2000
505 REM MONDE
510 DATA 4134,4549,5013,5545,6143,6798
515 REM PAYS INDUSTRIALISES
520 DATA 1131,1174,1224,1276,1327,1377
525 REM PAYS EN VOIE DE DEVELOPPEMENT
530 DATA 3003,3375,3789,4269,4816,5420
535 REM CHINE
540 DATA 978,1071,1151,1241,1348,1468
545 REM INDE
550 DATA 618,694,786,894,1013,1142
555 REM JAPON
560 DATA 112,117,122,127,131,135
565 REM USA
570 DATA 214,222,235,248,260,270
575 REM URSS
```

```
580 DATA 254,268,282,296,310,323
585 REM RFA
590 DATA 61.8,61.7,60.0,58.6,57.8,56.2
595 REM RDA
600 DATA 16.8,16.7,16.6,16.4,16.2,16.1
610 DATA "MONDE","PAYS I","PAYS E","CHINE","INDE"
620 DATA "JAPON","USA","URSS","RFA","RDA"
1000 REM ATTENDRE
1010 LOCATE 5,23:PRINT"PRESSER UNE TOUCHE S'IL VOUS PLAIT!"
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
1100 REM TAUX DE CROISSANCE
1110 R=EXP(LOG(Y/X)/25)-1
1120 R=R*100:R=INT(R*100+0.5)/100
1130 WR(L)=R
1140 RETURN
1200 REM TRAIT
1210 FOR I=1 TO 40:PRINT"-";:NEXT I
1220 PRINT:RETURN
```

5) Liste des variables

B = population
I = indice de boucle
J = année
L$ = pays
WR= taux de croissance
X = population initiale (1975) en millions d'habitants
Y = population finale (2000) en millions d'habitants

6) Description du programme

Ligne 10-60	: titre
Ligne 70-100	: lecture des données
Ligne 110	: saut dans le SP 1000 : attente
Ligne 140-145	: entête du tableau
Ligne 150	: saut dans le SP 1200 de soulignement
Ligne 160-190	: sortie des données connues
Ligne 200-230	: détermination du taux moyen de croissance annuelle pour les différents pays L =1, L =2 et L =3
Ligne 210	: initialisation des variables X et Y
Ligne 220	: saut dans le SP 1100 : calcul du taux de croissance
Ligne 240	: sortie des taux de croissance
Ligne 250	: pause
Ligne 260-390	: même procédure que celle décrite à la ligne 140-250 mais pour le deuxième tableau
Ligne 400-494	: idem pour le troisième tableau
Ligne 496	: fin du programme
Ligne 500-620	: données
Ligne 1000-1040	: sous-programme d'attente
Ligne 1100-1140	: sous-programme calculant le taux moyen de croissance annuel (il est déterminé à l'aide du calcul logarithmique, voir la ligne 1110)
Ligne 1200-1220	: sous-programme de trait : en ajoutant des signes moins les uns aux autres un trait est dessiné sur l'écran

7) Résultats

Ce programme élabore trois tableaux les uns au dessous des autres.

1er tableau (indications en millions)

ANNEE	MONDE	PAYS I	PAYS E
1975	4134	1131	3003
1980	4549	1174	3375
1985	5013	1224	3789
1990	5545	1276	4269
1995	6143	1327	4816
2000	6798	1377	5420
TAUX %	2.01	0.79	2.39

2ème tableau (indications en millions)

ANNEE	CHINE	INDE	JAPON	USA	URSS
1975	978	618	112	214	254
1980	1071	694	117	222	268
1985	1151	786	122	235	282
1990	1241	894	127	248	296
1995	1348	1013	131	260	310
2000	1468	1142	135	270	323
%	1.64	2.49	0.75	0.93	0.97

3ème tableau (indications en millions)

ANNNEE	RFA	RDA
1975	61.8	16.8
1980	61.7	16.7
1985	60	16.6
1990	58.6	16.4
1995	57.8	16.2
2000	56.2	16.1
%	-0.38	-0.17

ECONOMIE



8.1 Avant-Propos

L'économie et plus particulièrement l'industrie et le commerce, a longtemps été le principal domaine d'étude de l'informatique. C'est sans doute encore vrai aujourd'hui.

Les problèmes tels qu'ils se posent dans ces secteurs sont généralement formulés de manière relativement simple. Leur adaptation sur ordinateur, et aujourd'hui sur micro-ordinateur, en est facilité.

Les questions économiques les plus importantes, comportant des problèmes arithmétiques, sont généralement traitées en classe et notamment dans les écoles de commerce.

Nous allons étudier quelques-uns de ces problèmes dans ce chapitre.

Les problèmes sélectionnés ne figurent ici qu'à titre d'exemples.

8.2 Calcul des intérêts

1) Présentation du problème

Prenons l'exemple d'un individu qui place un certain capital dans une banque et dont les intérêts annuels sont variables.

Tous les ans le capital s'accroît selon un certain pourcentage. Il faut tenir compte du fait qu'à partir de la seconde année de placement les intérêts déjà acquis en rapportent d'autres (calcul des intérêts cumulés).

Nous allons écrire un programme qui calcule, en fonction de son montant initial, le montant d'un capital placé à un taux d'intérêts (%) pendant x années.

Ce programme peut d'ailleurs être pris comme modèle pour simuler différents phénomènes de croissance.

2) Analyse du problème

Appelons K0 le montant initial du capital, P(%) le taux d'intérêt et T la durée du placement (années).

A la fin de la première année, on peut calculer le montant du capital X1 comme suit :

X1 = X0 + X0 * P/100 = X0 * (1 + P/100)

A la fin de la seconde année, on obtient :

$$X2 = X1 + X1 * P/100 = X1 * (1 + P/100)$$

$$= X0 * (1 + P/100) * (1 + P/100)$$

$$= X0 * (1 + P/100)^2$$

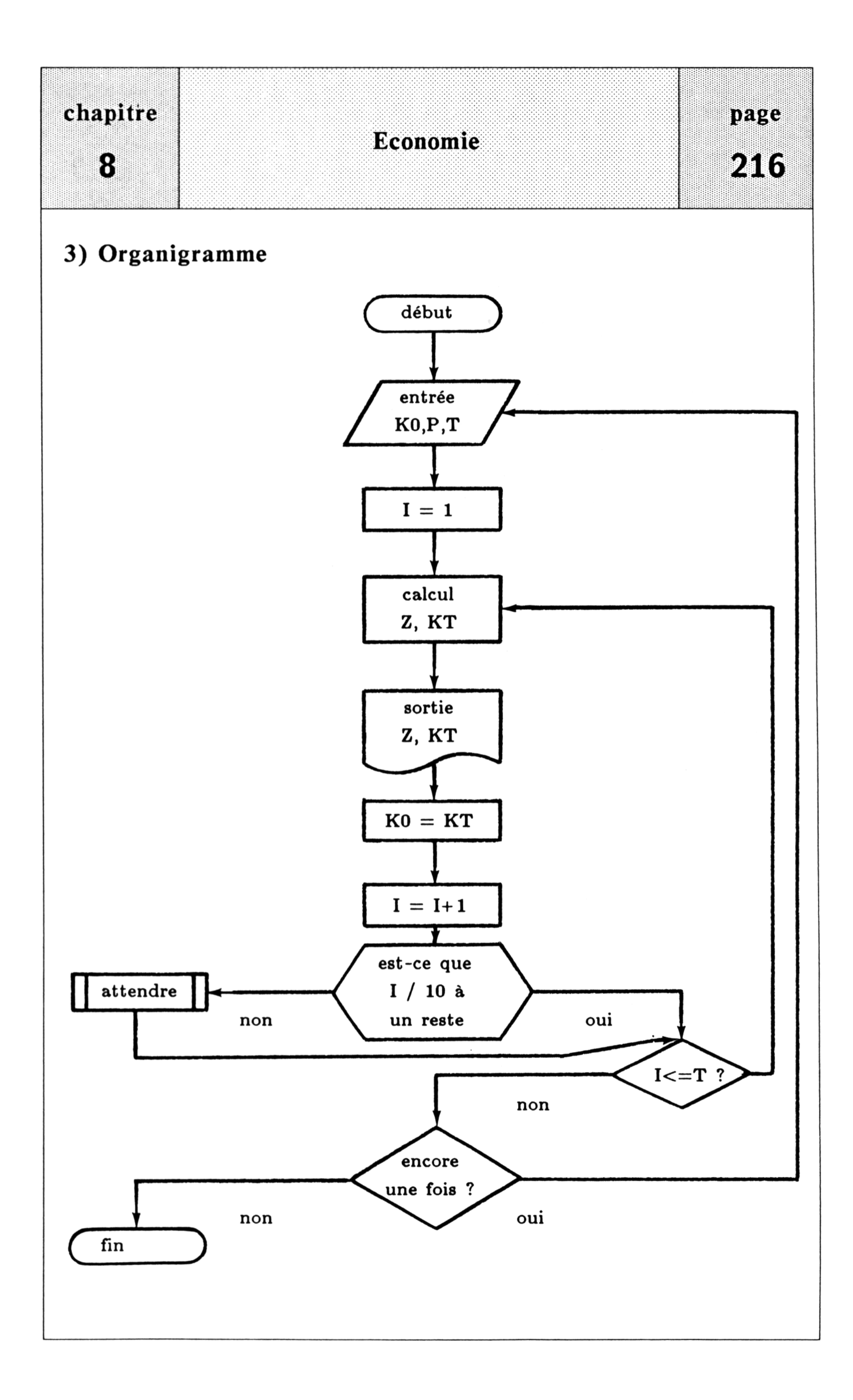
Notons que nous avons déjà rencontré ces équations dans l'un des chapitres précédents.

Après T années on obtient donc d'une manière générale :

$$XT = X0 * (1 + P/100)^T$$

Cette relation nous permet de construire le programme adéquat.

3) Organigramme



4) Programme

```
10 REM W1-CALCUL DES INTERETS
20 CLS
30 PRINT"PROGRAMME CALCULANT LES INTERETS POUR UN"
40 PRINT"PAIEMENT ANNUEL DES INTERETS ET UN CAPITAL"
50 PRINT"INITIAL DONNE"
60 PRINT:PRINT:PRINT"PROF.W.VOSS 1984"
70 PRINT:PRINT:PRINT:PRINT:PRINT
80 INPUT "CAPITAL INITIAL                : ";K0
90 PRINT:INPUT "TAUX D'INTERETS ANNUEL EN %   : ";P
100 PRINT:INPUT "NOMBRE D'ANNEE                : ";T
110 CLS
120 PRINT"ANNEE     INTERETS    CAPITAL":PRINT:PRINT
130 FOR I=1 TO T
150 Z=K0*P/100
160 KT=K0+Z
190 PRINT TAB (2) I;:PRINT TAB (12) USING "####.##";Z;:PRINT TAB (21)
 USING "######.##";KT
200 K0=KT
210 IF I/10=INT(I/10) THEN GOSUB 500 :REM ATTENDRE
220 NEXT I
230 PRINT:PRINT:PRINT"ENCORE UN CALCUL    (O/N) ";:INPUT A$
240 IF A$="O" THEN CLS:GOTO 80
250 PRINT:PRINT:PRINT"FIN DES CALCULS":END
500 REM ATTENDRE
510 LOCATE 5,23
520 PRINT"PRESSEZ UNE TOUCHE !"
530 A$=INKEY$:IF A$="" THEN 530
540 CLS:PRINT"ANNEE     INTERETS    CAPITAL":PRINT:PRINT
550 RETURN
```

5) Liste des variables

I = indice de boucle
K0= montant du capital initial
KT= montant du capital après x années de placement
P = taux d'intérêt annuel en pourcentage
T = durée du placement en années
Z = montant annuel des intérêts

6) Description du programme

Ligne 10-70 : titre
Ligne 80-100 : demande des données d'entrée
Ligne 110-120 : effacement de l'écran et sortie de l'entête du tableau de résultats
Ligne 130-220 : calculs :
150 : montant des intérêts annuels
160 : montant du capital avec les intérêts pour l'année en cours
190 : sortie
200 : K0 prend la valeur de KT pour le calcul du montant du capital et de ses intérêts l'année suivante
210 : tous les 10 ans interruption du programme par saut au SP 500
Ligne 230-250 : fin du programme à moins qu'un autre calcul ne soit souhaité (dans ce cas, retour à la ligne 80).
Ligne 500-550 : sous-programme d'attente, lorsque l'écran est saturé
510-520 : remarque
530 : attente jusqu'à la pression d'une touche
540 : effacement de l'écran et sortie réïtérée de l'entête du tableau
550 : fin du sous programme et retour en arrière

7) résultats

Entrons un capital initial de 100 FF, un taux d'intérêt de 5.5 % et une durée de 6 ans. Nous obtenons alors les résultats suivants :

ANNEE	INTERETS	CAPITAL
1	5.50	105.50
2	5.80	111.30
3	6.12	117.42
4	6.46	123.88
5	6.81	130.69
6	7.19	137.88

ENCORE UN CALCUL (O/N) ?

8.3 Amortissement d'un emprunt

1) Présentation du problème

Les problèmes liés aux remboursements de crédits ou aux hypothèques font partie des problèmes économiques les plus importants. Des programmes informatiques adéquats permettent de déterminer la durée de remboursement en fonction de différentes conditions admises au départ telles que le montant des emprunts, celui des intérêts et les modalités de remboursement.

Le programme qui suit calcule la durée d'amortissement en fonction de différents montants, taux d'intérêt et remboursements fixés au départ.

2) Analyse du problème

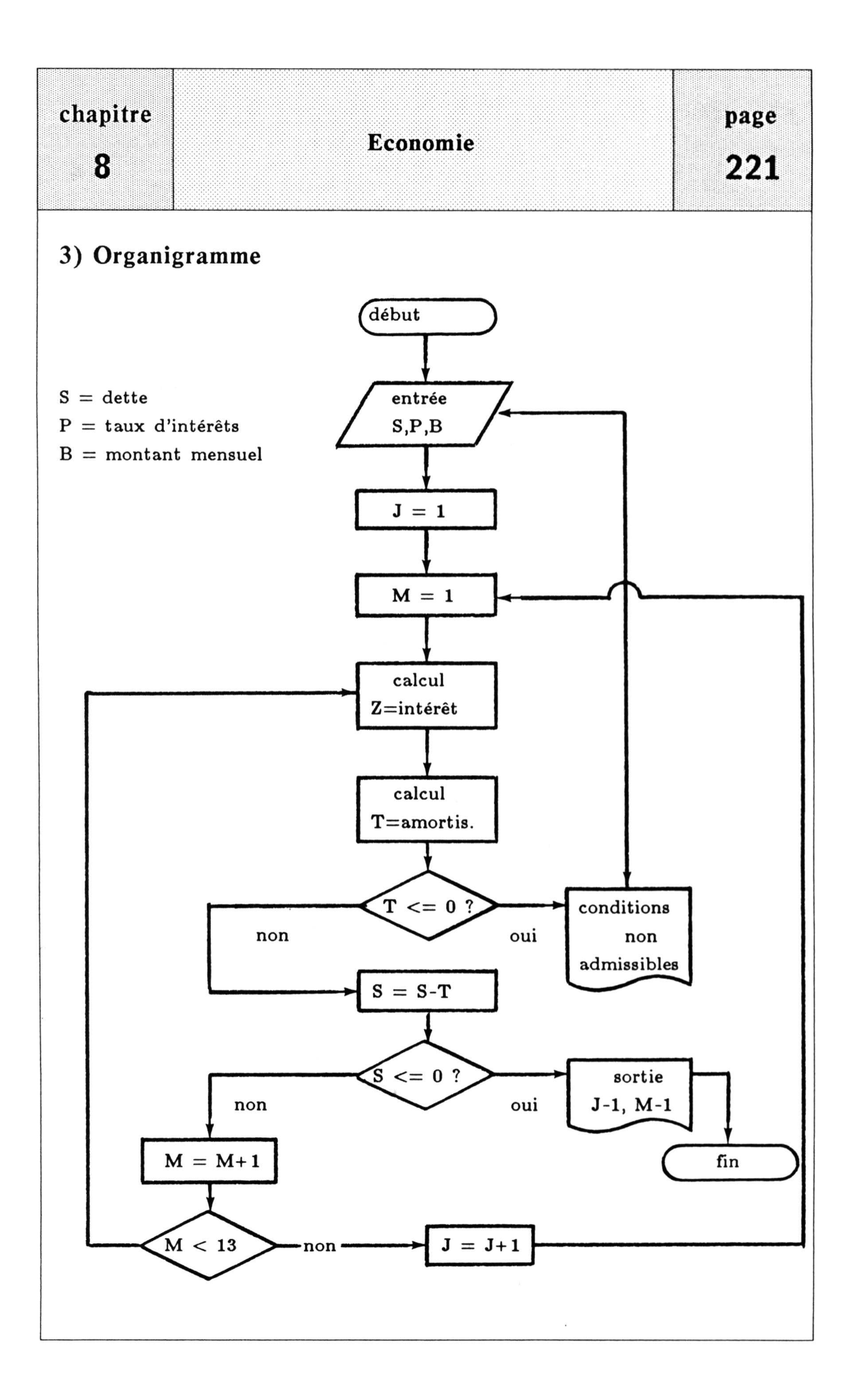
Prenons un exemple :

Un individu sollicite un emprunt d'une valeur de S Frs auprès d'une banque qui lui accorde un remboursement mensuel de B Frs. La mensualité est composée d'une partie remboursement des intérêts et d'une autre remboursement du capital. Ce dernier remboursement (obligatoirement positif) correspond à l'amortissement de la dette.

Le programme que l'on va étudier doit calculer les intérêts mensuels produits ainsi que le montant mensuel de l'amortissement. Si ce montant est négatif ou nul - c'est-à-dire que, mensuellement, les intérêts sont supérieurs au remboursement - le programme indique qu'il n'est pas possible, dans ces conditions d'amortir la dette.

Le montant de la dette est réactualisé chaque année. Dès que celui-ci devient négatif ou nul, la durée totale de l'amortissement de la dette est sortie.

3) Organigramme



4) Programme

```
10 REM W2-HYPOTHEQUE
20 CLS
30 PRINT"PROGRAMME POUR LE CALCUL DE LA DUREE D'UN AMORTISSEMENT"
40 PRINT TAB (10) "D'HYPOTHEQUE"
50 PRINT:PRINT:PRINT TAB(10)"PROF.W.VOSS 1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT:INPUT "MENSUEL                       : ";B
80 PRINT:INPUT "TAUX D'INTERET EN %           : ";P
90 PRINT:INPUT "DETTE D'HYPOTHEQUE ACTUELLE : ";S
100 CLS
110 PRINT" ANNEE    DETTE":PRINT
120 FOR I=1 TO 20:PRINT "-";:NEXT I:PRINT
130 J=1
140 M=1
150 Z=((P/100)*S)/12:T=B-Z
160 IF T <= 0 THEN 250
170 S=INT((S-T)*100+0.5)/100
180 IF S <= 0 THEN 300
190 M=M+1: IF M < 13 THEN 150
200 PRINT TAB (2)J;TAB (8)S
210 J=J+1:GOTO 140
250 PRINT:PRINT:PRINT"LES CONDITIONS SONT TELLES"
260 PRINT"QUE L'ON NE PEUX PAS AMORTIR"
270 PRINT:PRINT"UNE NOUVELLE ENTREE ":PRINT:GOTO 70
300 PRINT:PRINT:PRINT"DUREE DE L'AMORTISSEMENT : ":PRINT
310 PRINT J-1; "ANNEES  ";M-1;" MOIS."
320 PRINT:PRINT:PRINT "ENCORE UN CALCUL  (O/N) ?";:INPUT A$
330 IF A$="O" THEN CLS:GOTO 70
340 PRINT:PRINT:PRINT"FIN":END
```

5) Liste des variables

A$= variable chaîne de caractères (Oui, Non)
B = montant du remboursement mensuel
I = indice de boucle
J = indice de boucle (compteur des années)
M = indice de boucle (compteur des mois)
P = taux d'intérêt annuel de la dette
T = montant de l'amortissement mensuel
Z = montant des intérêts mensuel
S = montant de la dette

6) Description du programme

Ligne 10-60 : titre
Ligne 70-90 : demande des informations d'entrée
Ligne 100-120 : sortie de l'entête du tableau de résultats
Ligne 130-140 : initialisation du compteur d'années (J) et de mois M
Ligne 150-170 : calcul du montant des intérêts mensuels, de l'amortissement mensuel et de la nouvelle dette
Ligne 160 : si l'on ne peut pas amortir, poursuite à la ligne 250
Ligne 180 : si la dette est amortie poursuite à la ligne 300
Ligne 190 : mois suivant si M est inférieur à 13, retour à la ligne 150, sinon poursuite à la ligne 200
Ligne 200 : sortie de la dette du moment
Ligne 210 : année suivante et poursuite en 140
Ligne 250-270 : message à l'écran si les conditions d'amortissement sont acceptables et retour à la ligne 70

Ligne 300-310 : sortie de la durée d'amortissement en années et en mois (le lecteur doit comprendre pourquoi la valeur 1 doit être enlevée de J et de M)

Ligne 320-340 : fin du programme à moins qu'un nouveau calcul ne soit souhaité (dans ce cas, après effacement de l'écran, retour à la ligne 70)

7) Résultats

Si nous entrons, pour répondre aux questions du programme, les informations suivantes :

```
TAUX MENSUEL                 : ? 500
TAUX D'INTERET EN %          : ? 6.5
HYP ACTUELLE : DETTE         : ? 40000
```

Lors du déroulement du programme nous obtenons :

```
ANNEE              DETTE

1                  36 496.86
2                  32 759.09
3                  28 771.02
4                  24 515.85
5                  19 975.70
6                  15 131.47
7                   9 962.84
8                   4 448.06

DUREE D'AMORTISSEMENT :

8 ANS ET 9 MOIS

ENCORE UN CALCUL (O/N) ?
```

8.4 Moyenne arithmétique

1) Présentation du problème

Il est courant en économie d'avoir recours au calcul de moyennes arithmétiques.

Le programme étudié ici effectue ce calcul à partir d'un nombre quelconque de données préalablement saisies.

2) Analyse du problème

L'analyse du problème ne crée pas de difficulté particulière.

Comme l'on veut pouvoir traiter un nombre quelconque de données il faut tout d'abord communiquer à l'ordinateur le nombre désiré, à chaque début de traitement.

Les valeurs peuvent être immédiatement additionnées les unes aux autres dès leur saisie (la moyenne arithmétique est bien la somme de toutes les valeurs divisée par le nombre total de ces valeurs). Le résultat de la division de la somme obtenue par le nombre de valeurs est affiché.

Si un autre calcul doit être exécuté, le programme revient au début, sinon il est terminé.

3) Organigramme

4) Programme

```
10 REM W3-MOYENNE
20 CLS
30 PRINT"PROGRAMME POUR CALCULER UNE MOYENNE"
40 PRINT TAB(11)"ARITHMETIQUE."
50 PRINT:PRINT:PRINT TAB(9)"PROF.  W.VOSS,1984"
60 PRINT:PRINT:PRINT:PRINT:PRINT
70 INPUT"COMBIEN DE VALEURS:";N
80 CLS:S=0
90 FOR I=1 TO N
100 PRINT I;".VALEUR: ";:INPUT X
110 S=S+X
120 NEXT I
130 AM=S/N
150 PRINT:PRINT:PRINT"MOYENNE    ";AM:PRINT
160 PRINT:PRINT"ENCORE UN CALCUL  ? (O/N)";:INPUT A$
170 IF A$="O" THEN CLS:GOTO 70
180 PRINT:PRINT:PRINT"FIN DES CALCULS  ":END
```

5) Liste des variables

AM = moyenne arithmétique
A$= variable chaîne de caractères (Oui, Non)
I = indice de boucle
N = nombre de valeurs à prendre en compte pour le calcul de la moyenne
S = somme des valeurs
X = valeur qui entre dans le calcul de la moyenne

6) Description du programme

Ligne 10-60 : titre
Ligne 70 : indication du nombre des valeurs à prendre en compte pour le calcul de la moyenne
Ligne 80 : effacement de l'écran et initialisation de la variable S
Ligne 90-120 : entrée et addition des valeurs les unes après les autres
Ligne 130-150 : calcul de la moyenne arithmétique et sortie du résultat
Ligne 160-180 : fin du programme à moins qu'un calcul supplémentaire ne soit souhaité.

7) Résultats

Si nous indiquons qu'il y a, par exemple, 3 valeurs à traiter, le programme en réclame bien évidemment trois.

Si, par exemple, ces valeurs sont 5, 7 et 9, nous obtenons :

```
MOYENNE = 7
ENCORE UN CALCUL (O/N) ?
```

8.5 Répartition de fréquence

1) Présentation du problème

Pour l'exploitation de grands stocks de données (l'ordinateur y excelle) il existe une première étape statistique qui consiste en la production de ce que l'on appelle la répartition de fréquence.

Il s'agit là d'attribuer aux valeurs d'indice (ou classes de valeur d'indice) une variable d'enquête, (la fréquence de leur apparition), ceci au moyen d'un graphique ou d'un tableau.

Le programme établira la répartition de fréquence sous la forme d'un graphique et d'un tableau. Nous prendrons comme données de travail la répartition des âges en France pour l'année 1980.

On peut représenter cette répartition comme suit :

Répartition des âges en France en 1980 :

```
AGE                    NOMBRE EN %

------------------------------------
 0   JUSQU'A  10          13.1
10   JUSQU'A  20          16.1
20   JUSQU'A  30          14.9
30   JUSQU'A  40          13.1
40   JUSQU'A  50          12.1
50   JUSQU'A  60          11.9
60   JUSQU'A  70           8.3
70   JUSQU'A  80           6.8
80   JUSQU'A  90           3.2
90   JUSQU'A 100           0.2
```

2) Analyse du problème

Pour analyser le problème, nous partons du fait que les données nous sont présentées de telle manière que celles-ci puissent être entrées telles quelles comme matériel d'entrée du programme.

Si tel n'était pas le cas, si par exemple on avait prélevé dans un sondage, ou ailleurs, des indications individuelles d'âge, il faudrait d'abord veiller, dans une première partie du programme, à ce que les réponses individuelles soient immédiatement réparties dans les différentes classes d'âge disponibles (ceci peut être obtenu par une série d'instructions IF...THEN où chaque âge est contrôlé).

Après cela, les nombres obtenus occupant les classes pourraient être relativisés en les rapportant à l'ensemble des informations pour obtenir des pourcentages semblables à ceux qui sont présentés dans le tableau ci-dessus.

Cette partie du programme pourrait ressembler à ce schéma :

```
100 INPUT " ages : ";X
110 IF X < 10 THEN Z1=Z1+1 : GOTO 100
120 IF X < 20 THEN Z2=Z2+1 : GOTO 100
...

...

200 R1=(Z1/N) * 100
210 R2=(Z2/N) * 100
...

...
```

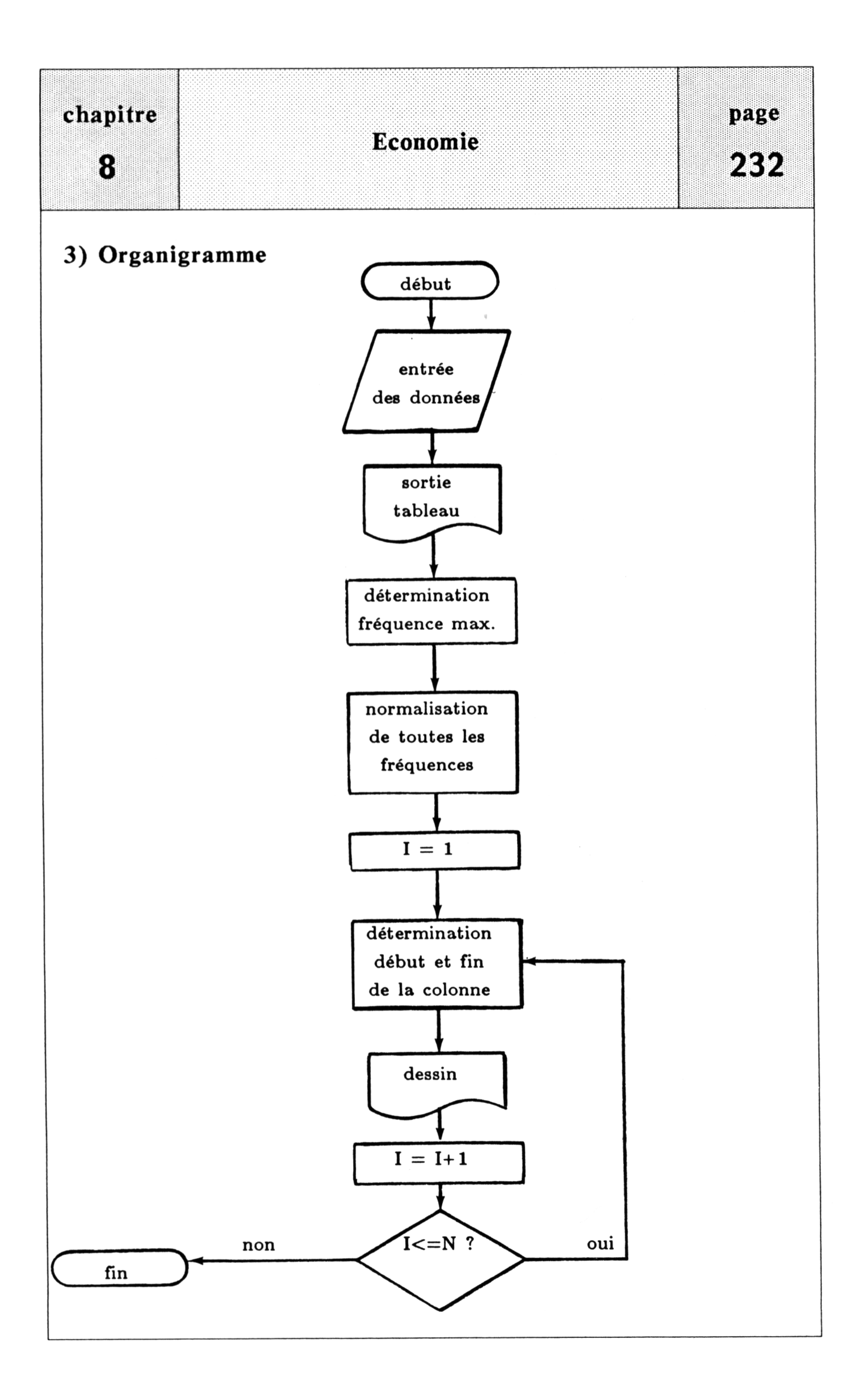
Dans cet exemple il s'agit de représenter, dans un premier temps, les valeurs données sous la forme d'un tableau. Cela n'entraine aucune difficulté particulière de programmation.

La représentation graphique de cette répartition de fréquence est, par contre, un peu moins aisée.

Pour pouvoir utiliser l'écran le mieux possible au cours de la représentation graphique nous déterminons tout d'abord la plus grande fréquence à partir du stock de données que nous possédons. Nous calculons ensuite toutes les autres fréquences de manière à ce que la plus grande tienne verticalement sur 20 lignes de l'écran.

Nous affichons les fréquences comme étant des colonnes verticales.
Pour obtenir un dessin disposé en colonnes, il est nécessaire de déterminer leur point de départ et leur extrémité finale. Nous obtenons donc à chaque fois deux positions sur l'écran. Il ne s'agit plus maintenant que d'afficher ces deux valeurs et toutes les positions intermédiaires.

3) Organigramme



4) Programme

```
10 REM W5-REPARTITION
20 CLS
30 PRINT"PROGRAMME POUR LA REPRESENTATION PAR UN TABLEAU ET"
40 PRINT"UN SCHEMA D'UNE REPARTITION"
50 PRINT"STATISTIQUE DE FREQUENCE."
60 PRINT:PRINT:PRINT TAB (10) "PROF.W.VOSS 1984"
70 PRINT:PRINT:PRINT:PRINT:PRINT
80 PRINT"ON REPRESENTE LA REPARTITION DES AGES"
90 PRINT"EN FRANCE POUR L'ANNEE 1980"
110 GOSUB 1000:REM ATTENDRE
120 DIM F(10),FT(10)
130 FOR I=1 TO 10:READ F(I):NEXT I
140 PRINT"AGE ";TAB(22) "NOMBRE EN %":PRINT
150 FOR I=1 TO 35:PRINT"-";:NEXT I:PRINT
160 FOR I=1 TO 10
165 A=(I-1)*10:B=I*10
170 PRINT USING "##";A;
172 PRINT TAB(6) "JUSQU'A ";
174 PRINT USING "###";B;
176 PRINT TAB(24) USING "###.#";F(I)
180 NEXT I
190 GOSUB 1000
220 REM NORMALISATION
230 FM=0
240 FOR I=1 TO 10:IF F(I)>FM THEN FM=F(I)
250 NEXT I
260 FOR I=1 TO 10
270 FT(I)=F(I)*(20/FM)
280 NEXT I
290 REM DESSIN
300 B=20
310 FOR I=1 TO 10
320 A=21-FT(I):A=INT(A+0.5)
330 FOR Z=A TO B
340 LOCATE 4*I,Z
```

```
350 PRINT CHR$(143)
360 NEXT Z
370 NEXT I
380 PLOT 35,80:DRAW 630,80
390 PRINT:PRINT TAB(4) "AGE PAR CLASSE DE 10 ANS"
430 PRINT"FIN":END
500 DATA 13.1,16.1,14.9,13.1,12.1,11.9,8.3,6.8,3.2,.2
1000 REM ATTENDRE
1010 LOCATE 5,23:PRINT"PRESSEZ UNE TOUCHE  S.V.P"
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
```

5) Liste des variables

A = début de la classe d'âge :
adresse de départ de la colonne à dessiner
A$= variable chaine de caractères pour la réception des réponses
B = fin de la classe d'âge :
adresse finale de la colonne à dessiner
F = fréquence
FM= fréquence maximale
FT= tableau à une dimension des fréquences transformées
I = indice de boucle
Z = indice de boucle (ligne de l'écran)

6) Description du programme

Ligne 10-90 : adresse
Ligne 110 : attente
Ligne 120-130 : dimensionnement et lecture des données de départ
Ligne 140-150 : sortie de l'adresse du tableau
Ligne 160-180 : sortie du tableau
165 : détermination du point de départ des classes et du point final des classes
170-176 : sortie
Ligne 190 : attente
Ligne 220-250 : recherche de la plus grande fréquence en comparant toutes les fréquences avec la plus grande obtenue (en commençant par la valeur zéro)
Ligne 260-280 : détermination de la fréquence transformée par l'obligation des fréquences sur la fréquence maximale
Ligne 290-370 : dessin de 10 colonnes :
300 : fin de la colonne
320 : détermination du début de la colonne d'après la répartition des lignes de l'écran
330-360 : dessin de colonnes par l'ajout de petits carrés dans la colonne 4 * I
Ligne 380 : axe horizontal
Ligne 390 : inscription de l'axe horizontal
Ligne 430 : fin du programme
Ligne 500 : données
Ligne 1000-1040 : sous-programme d'attente

7) Résultats

Nous n'allons pas présenter les résultats de ce programme. Il est bien plus utile pour le lecteur de comparer l'affichage des données sur l'écran avec la description faite au début de ce paragraphe.

8.6 Puissance économique des régions françaises

1) Présentation du problème

Le programme suivant est d'un genre tout différent de celui qui précède. Il s'agit à nouveau ici, comme dans quelques paragraphes antérieurs, de fabriquer un "programme d'information".

Nous allons développer un programme qui sorte des données économiques importantes concernant les régions françaises, et ce à la demande de l'utilisateur.

Les données devant être présentées pour cette mise à disposition de l'information sont les suivantes :

1. densité de la population française (en millions)
2. superficie (en milliers de kilomètres carrés)
3. produit national brut (en milliards de FF)
4. nombre de salariés
5. taux de salariés étrangers (en %)
6. produit total des impôts (en milliard de FF)
7. dettes (en milliards de FF)
8. part de la production industrielle (en %)

Il va de soi que par besoins où intérêts particuliers cette liste pourrait être allongée ou que l'on pourrait envisager d'autres thèmes que les régions. Il faudrait naturellement modifier le stock de données en conséquence.

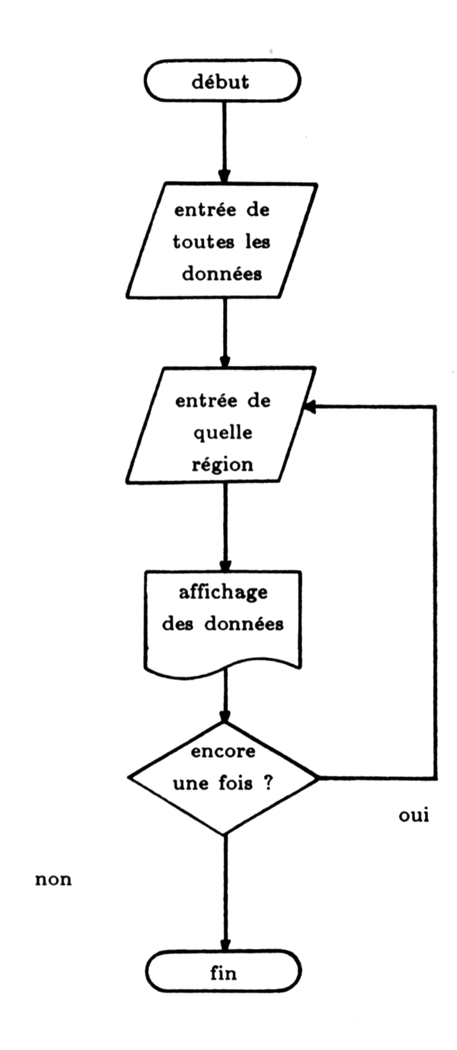
2) Analyse du problème

Le lecteur aura déjà remarqué que dans les différents chapitres certains types de problèmes se répètent ou reviennent sous des formes analogues.

Il en va ainsi de cet énoncé : il ne pose, en principe, pas de problème nouveau si bien que l'analyse est très simple :

Il faut en premier lieu entrer toutes les informations qui nous intéressent. On demande ensuite à l'utilisateur d'indiquer la région sur laquelle il souhaite avoir des informations. C'est seulement après cette entrée que se produit l'affichage des données sur la région souhaitée.

3) Organigramme



4) Programme

```
10 REM W6-REGIONS
20 CLS
30 PRINT"PROGRAMME POUR LA SORTIE D'INFORMATIONS SUR LA"
40 PRINT"PUISSANCE ECONOMIQUE DES REGIONS"
50 PRINT
60 PRINT:PRINT:PRINT TAB(10) "PROF.W.VOSS 1984"
70 PRINT:PRINT:PRINT:PRINT:PRINT
110 DIM L$(22),V(8,22),B$(8)
120 FOR I=1 TO 22: READ L$(I):NEXT I
130 FOR I=1 TO 8: READ B$(I):NEXT I
140 FOR J=1 TO 8:FOR I=1 TO 12:READ V(J,I):NEXT I:NEXT J
150 GOSUB 1000:REM ATTENDRE
200 PRINT"LES INFORMATIONS SONT SOUHAITES POUR"
210 PRINT"QUELLE REGION  ?":PRINT
220 FOR I=1 TO 22
230 PRINT TAB(5) "(";I;")";:PRINT TAB (13)L$(I)
240 NEXT I
250 PRINT:PRINT:INPUT "INDIQUEZ LE NUMERO  S.V.P :";Z
260 CLS
280 PRINT L$(Z):PRINT:PRINT
290 FOR J=1 TO 8
300 PRINT B$(J); TAB(39)": ";:PRINT V(J,Z)
310 NEXT J
320 GOSUB 1000:REM ATTENDRE
330 PRINT "ENCORE UN CALCUL   (O/N) ";:INPUT A$
340 IF A$="O" THEN CLS:GOTO 200
350 PRINT:PRINT"FIN":END
500 DATA NORD-PAS DE CALAIS,PICARDIE,HAUTE NORMANDIE,BASSE NORMANDIE,C
HAMPAGNE-ARDENNES,LORRAINE,ALSACE
510 DATA REGION PARISIENNE,BRETAGNE,PAYS DE LOIRE,CENTRE,BOURGOGNE,FRA
NCHE COMTE,POITOU-CHARENTES
520 DATA LIMOUSIN,AUVERGNE,RHONES-ALPES,AQUITAINE,MIDI-PYRENEES,LANGUE
DOC-ROUSSILLON,PROVENCE-COTE D'AZUR,CORSE
550 DATA POPULATION,SUPERFICIE,PRODUIT NATIONAL,SALARIE
560 DATA NOMBRE D'ETRANGERS,PRODUIT DES IMPOTS,DETTES,PART DE PRODUCTI
```

```
ON INDUSTRIELLE
600 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
605 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
610 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
615 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
620 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
625 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1000 REM ATTENDRE
1010 LOCATE 5,23:PRINT"PRESSEZ UNE TOUCHE  S.V.P"
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
```

5) Liste des variables

A$= variable chaîne de caractères

B$ = désignations des variables

I = indice de boucle (régions)

J = indice de boucle

L$ = noms des régions

V = variables

Z = numéro de la région souhaitée

6) Description du programme

Ligne 10-70 : adresse
Ligne 110 : dimensionnement
Ligne 120-140 : lecture des données de départ :
  120 : noms des régions
  130 : désignations des variables
  140 : valeurs des variables
Ligne 150 : attente
Ligne 200-250 : réclamation du numéro de la région sur laquelle des informations sont souhaitées, après qu'une liste des régions ait été présentée sur l'écran
Ligne 260-280 : effacement d'écran et sortie du nom de la région demandée
Ligne 290-310 : sortie de l'enregistrement de données sur la région souhaitée
Ligne 320 : attente
Ligne 330-350 : demande pour savoir si une autre sortie est souhaitée (si oui, effacement d'écran et retour à l'enregistrement 200), si non, fin du programme
Ligne 500-506 : noms des régions dans les DATAs
Ligne 510-514 : désignations des variables dans les DATAs
Ligne 600-618 : valeurs des variables : il faut indiquer les valeurs de manière à ce que pour la première variable toutes les valeurs des régions suivent, ainsi que pour la deuxième etc... (voir l'enregistrement 140)

ATTENTION : Il n'y a que des zéros dans le programme (voir l'étape suivante !)

Ligne
1000-1040 : sous-programme en attente

7) résultats

Il n'y a que des zéros dans le stock de données du programme présenté si bien qu'en réalité seules les valeurs nulles pourraient apparaitre sur l'écran au cours d'un essai de fonctionnement.

Si un essai de fonctionnement devait être accompli "pour de vrai", il faudrait d'abord entrer les données concrètes.

MATHEMATIQUES II



9.1 Avant-Propos

Voici pour conclure un chapitre dans lequel d'autres problèmes mathématiques sont traités, et en particulier ceux de degré de difficulté plus élevé qu'auparavant. Ceci pour que l'utilisateur ayant un bon niveau puisse se rendre compte qu'il est possible d'aborder des problèmes plus complexes avec des programmes assez simples.

Il se peut donc que ce chapitre soit un petit peu difficile pour un débutant mais c'est justement ce qui fait son intérêt.

Ces problèmes sont programmés sans utiliser de nouvelles instructions BASIC. Ceci implique que les programmes construits ne peuvent pas répondre au souci d'élégance que peuvent avoir d'autres programmes, Notre but étant avant tout la simplicité de compréhension des programmes.

Nous détaillerons beaucoup moins chaque exemple que dans les chapîtres précédents.

Après un bref énoncé du problème et quelques remarques pour son analyse, nous présenterons immédiatement le programme et sa description rapide. Les étapes "organigramme" et "résultats" ne figurent plus par la suite, il n'y a aucun élément nouveau en ce qui concerne la logique de programmation.

9.2 Réseaux

Enoncé du problème

Ce programme permet la sortie à l'écran d'un réseau de lignes ayant un écart quelconque.

Ce programme simple peut être utilisé pour, par exemple, séparer par des traits horizontaux et verticaux d'importantes données dans un tableaux.

Analyse du problème

Cette analyse est très simple : l'utilisateur doit donner des indications sur l'écart séparant les lignes et les colonnes. Conformément à ces indications, le curseur graphique est positionné en un point (instruction PLOT) avant le tracé d'une ligne droite (instruction DRAW).

Programme

```
10 REM Z2-GRILLE
20 CLS
30 PRINT"PROGRAMME POUR  DESSINER UNE GRILLE"
40 PRINT:PRINT:PRINT TAB(9)"PROF.  W.VOSS, 1984"
50 LOCATE 3,20:INPUT"ECART DE COLONNE : ";S
60 LOCATE 3,22:INPUT"ECART DE LIGNE : ";Z
70 CLS
80 FOR I=1 TO 639 STEP S
90 PLOT I,399:DRAW I,1
100 NEXT I
110 FOR I=1 TO 399 STEP Z
120 PLOT 1,I:DRAW 639,I
130 NEXT I
140 LOCATE 18,24:PRINT"FIN":END
```

Liste des variables

I = indice de boucle

S = écart entre les colonnes

Z = écart entre les lignes

Description du programme

Ligne 10-40	: titre
Ligne 50-60	: entrée des écarts entre les colonnes et entre les lignes
Ligne 70	: effacement de l'écran
Ligne 80-100	: tracé des lignes verticales
Ligne 110-130	: tracé des lignes horizontales
Ligne 140	: fin du programme

9.3 La droite

Enoncé du problème

Le programme permet, dans un système de coordonnées dont l'origine se trouve au centre de l'écran (320, 200), de tracer une droite quelle que soit son équation.

Analyse du problème

La position d'une droite est déterminée par deux paramètres : son ordonnée initiale (A) et la tangente de son angle d'inclinaison (B) (coefficient directeur de la droite), soit l'équation suivante :

$$Y_i = A + B * X_i$$

Comme l'origine des coordonnées doit se trouver au point (320, 200), il s'ensuit que :

$$Y_i = 200 + A + B *(X_i - 320)$$

Programme

```
10 REM Z2-DROITE
20 CLS
30 PRINT"PROGRAMME POUR LE DESSIN D'UNE DROITE."
40 PRINT:PRINT:PRINT TAB(9)"PROF.  W.VOSS, 1984"
50 LOCATE 3,20:INPUT"SEGMENT DES ORDONNEES A : ";A
60 LOCATE 3,22:INPUT"PENTE B :";B
70 CLS
80 PLOT 0,200:DRAW 639,200
90 PLOT 320,399:DRAW 320,0
100 FOR X=0 TO 639
110 Y=200+A+B*(X-320)
120 IF Y<=0 OR Y>=399 THEN 140
130 PLOT X,Y
140 NEXT X
150 LOCATE 18,24:PRINT"FIN":END
```

Liste des variables

A = ordonnée initiale

B = tangente de l'angle (entre la droite et l'horizontale passant par A) ou coefficient directeur de la droite

X = valeurs sur l'axe des abscisses

Y = valeurs de la fonction

Description du programme

Ligne 10-40 : titre
Ligne 50-60 : entrée des paramètres A et B
Ligne 70 : effacement de l'écran
Ligne 80-90 : tracé du système de coordonnées
Ligne 100-140 : tracé de la droite sur toute la largeur de l'écran; pour les valeurs non représentables, on saute l'instruction 130
Ligne 150 : fin du programme

9.4 Le cercle

Enoncé du problème

D'après le même modèle que celui de la section précédente nous présentons un programme destiné à tracer un cercle quel qu'il soit.

Le programme est analogue au précédent sur bien des points, ce qui nous permet d'être brefs.

Analyse du problème

Nous nous trouvons ici devant le même problème que celui de la section précédente avec bien sûr le fait que nous partons maintenant de l'équation d'un cercle :

$$(x - a)^2 + (y - b)^2 = r^2$$

Où **r** représente le rayon du cercle, **x** et **y** sont les coordonnées de la circonférence et **a** et **b** les coordonnées du centre du cercle.

Si l'on résoud cette équation du second degré, on obtient les équations des demi-cercles supérieur et inférieur.

C'est à dire :

$$(y - b)^2 = r^2 - (x - a)^2 = D$$

On en déduit :

$$y_1 = b + SQR(D)$$

$$y_2 = b - SQR(D)$$

Nous connaissons maintenant les éléments nécessaires pour traiter le programme présenté :

Ses données sont les coordonnées du point central (a et b appellés par la suite T et Z) et le rayon du cercle r (R dans le programme). On peut déterminer ensuite les valeurs y1 et y2 correspondant aux différentes valeurs de X à partir de la relation donnée ci-dessus. Les points (X, y1) et (X, y2) peuvent alors être représentés.

Il faut remarquer à ce propos, que X ne peut varier que de T-R à T+R. En dehors de cette zone aucune valeur réelle de Y n'est définie (la grandeur auxiliaire D serait alors négative et l'extraction de sa racine carrée mènerait à une erreur et à une interruption de programme).

Programme

```
10 REM Z3-CERCLE
20 CLS
30 PRINT"PROGRAMME POUR DESSINER UN CERCLE"
40 PRINT:PRINT:PRINT:PRINT:PRINT
50 PRINT"CE PROGRAMME REQUIERT COMME INFORMATIONS"
60 PRINT"LES DONNEES ET LES LIGNES DU POINT"
70 PRINT"CENTRAL ET LE RAYON DU CERCLE"
80 PRINT
90 PRINT:PRINT:PRINT:INPUT"CENTRE-DONNEES";T
100 PRINT:INPUT"CENTRE-LIGNE:";Z
110 PRINT:PRINT:INPUT "RAYON: ";R
120 CLS
130 PLOT 0,0:DRAW 0,399
140 PLOT 0,0:DRAW 639,0
150 PLOT T,0:DRAW T,399
160 PLOT 0,Z:DRAW 639,Z
170 A=T-R:B=T+R
180 IF A<0 THEN A=0
190 IF B>639 THEN B=639
200 FOR X=A TO B
210 D=R*R-(X-T)^2
220 IF D<=0 THEN Y=Z:GOTO 250
230 Y= Z+SQR(D)
240 IF Y<0 OR Y>399 THEN 260
250 PLOT X,Y
260 NEXT X
270 FOR X=B TO A STEP-1
280 D=R*R-(X-T)^2
290 IF D<=0 THEN Y=Z:GOTO 320
300 Y=Z-SQR(D)
310 IF Y<0 OR Y>399 THEN 330
320 PLOT X,Y
330 NEXT X
340 LOCATE 18,24:PRINT"FIN":END
```

Liste des variables

A = point correspondant à l'extrémité gauche du cercle
B = point correspondant à l'extrémité droite du cercle
D = $R^2 - (X-T)^2$
R = rayon
T = abscisse du centre du cercle (colonne)
X = coordonnées de X (abscisses)
Y = coordonnées de Y (ordonnées)
Z = ordonnée du centre du cercle (ligne)

Description du programme

Ligne 10-80 : titre et explications
Ligne 90-120 : données d'entrée et effacement de l'écran
Ligne 130-140 : tracé du système de coordonnées de l'écran
Ligne 150-160 : tracé du système de coordonnées du cercle
Ligne 170-190 : détermination de l'intervalle de valeurs de X
Ligne 200-260 : dessin du demi-cercle supérieur
Ligne 270-330 : dessin du demi-cercle inférieur
Ligne 340 : fin du programme

9.5 La sinusoïde

Enoncé du problème

Dans ce dernier chapitre portant sur le graphisme haute résolution nous présentons une oscillation sinusoïdale déjà rencontrée dans le chapitre intitulé "physique". La manière dont nous allons procéder est semblable à celle des deux sections précédentes.

Analyse du problème

L'analyse est ici très simple, en effet aucun problème nouveau ne vient se poser. Il nous suffit de veiller à ce que la périodicité de la ligne sinusoïdale (fréquence) et son amplitude (déplacement) soient telles que l'oscillation occupe l'écran de façon maximale.

Programme

```
10 REM Z4-SINUS
20 CLS
30 PRINT"PROGRAMME POUR LE DESSIN D'UNE DROITE SINUSOIDALE."
40 PRINT:PRINT:PRINT:PRINT TAB(9)"PROF.W.VOSS, 1984"
50 PRINT:PRINT:PRINT:PRINT:
60 PRINT"CE PROGRAMME REQUIERT LES INFORMATIONS
70 PRINT"INPUT SUIVANTES:":PRINT:PRINT
80 PRINT:INPUT"FREQUENCE EN UNITES DE PI";F
90 PRINT:INPUT"AMPLITUDE (0 J'USQUA 199)";A
100 CLS
110 PLOT 0,200:DRAW 639,200
120 FOR X=0 TO 639
130 J=X/100:J=J*(1/F)
140 Y=SIN(J):Y=Y*A
150 Y=200+Y
160 PLOT X,Y
170 NEXT X
172 LOCATE 19,12:PRINT"PI"
174 LOCATE 38,12:PRINT"2PI"
176 LOCATE 3,21:PRINT"FREQUENCE(nPI) = ";F
178 LOCATE 3,22:PRINT"AMPLITUDE    = ";A
180 LOCATE 18,24:PRINT"FIN":END
```

Liste des variables

A = amplitude
J = valeur normalisée de X
F = fréquence
X = coordonnée de X
Y = coordonnée de Y

Description du programme

ligne 10-50 : titre
ligne 60-90 : données d'entrée
ligne 100-110 : effacement de l'écran et tracé d'une ligne médiane horizontale
ligne 120-170 : détermination de la valeur du sinus et représentation, de cette valeur (après transformation d'après F et A)
ligne 172-180 : sortie du texte et fin du programme

9.6 Tableau numérique 1

Enoncé du problème

Ce programme calcule pour tous les nombres naturels de 1 à 100, leur carré et leur racine carrée.

Il faut donc considérer ce programme comme un "programme de consultation".

Analyse du problème

L'énoncé du problème est ici si clair qu'il est inutile d'en présenter l'analyse.

Programme

```
10 REM Z5-TABLEAU 1
20 CLS
30 PRINT"PROGRAMME POUR L'EXTRACTION DES CARRES"
40 PRINT"ET DES RACINES CARREES"
45 PRINT TAB(7)"DES NOMBRES DE 1 A 100"
50 PRINT:PRINT:PRINT TAB(9)"PROF.W.VOSS, 1984"
60 GOSUB 2000:REM ATTENDRE
70 FOR I=1 TO 100
80 Q=I*I:W=SQR(I)
90 PRINT USING "###";I;: PRINT TAB(7) USING "#####";Q;:PRINT USING "##
#.###";W

100 IF I/15=INT(I/15) THEN GOSUB 2000:REM ATTENDRE
110 NEXT I
120 PRINT:PRINT:END
1000 REM UP EN-TETE
1010 PRINT"NOMBRE CARRE RACINE":PRINT
1020 FOR J=1 TO 23:PRINT"-";:NEXT J:PRINT:RETURN
2000 REM UP ATTENDRE
2010 LOCATE 5,23:PRINT"PRESSER UNE TOUCHE S'IL VOUS PLAIT!"
2020 A$ =INKEY$:IF A$="" THEN 2020
2030 CLS
2040 GOSUB 1000: REM EN-TETE
2050 RETURN
```

Liste des variables

A$= variable chaine de caractères de réponse
I = indice de boucle (nombres naturels de 1 à 100)
J = indice de boucle
Q = carré du nombre traité
W = racine carrée du nombre traité

Description du programme

ligne 10-50	: titre
ligne 60	: appel du sous-programme d'attente (2000)
ligne 70-100	: calcul et sortie des valeurs concernées
ligne 100	: toutes les 15 lignes appel du sous-programme d'attente (2000)
ligne 120	: fin du programme
ligne 1000-1020	: sous-programme pour la sortie de l'en-tête du tableau
ligne 2000-2050	: sous-programme d'attente
ligne 2010	: sortie d'un message
ligne 2020	: attente de la frappe d'une touche
ligne 2030-2050	: après frappe de la touche effacement de l'écran, sortie d'un nouvel en-tête du tableau par un appel du SP 1000 puis retour au programme principal

9.7 Tableau numérique 2

Enoncé du problème

Ce programme permet la sortie, sous la forme d'un tableau, des fonctions trigonométriques suivantes :

- sinus (SIN)
- cosinus (COS)
- tangente (TAN)

Les résultats de ces fonctions sont affichés à partir d'angles exprimés en radians (unité du paramètre II, appelé PI dans le programme). Les angles sont xprimés en degrés dans le tableau.

Analyse du problème

Ici non plus l'analyse n'est pas nécessaire.

Programme

```
10 REM Z6-TABLEAU 2
20 CLS
30 PRINT TAB(11)"TABLEAU NUMERIQUE 2":PRINT:PRINT
40 PRINT"PROGRAMME POUR DETERMINER LES VALEURS DES"
50 PRINT"FONCTIONS ANGULAIRES":PRINT
60 PRINT TAB(11)"SINUS    (SIN)"
70 PRINT TAB(11)"COSINUS  (COS)"
80 PRINT TAB(11)"TANGENTE (TAN)"
85 PRINT:PRINT
90 GOSUB 2000:REM ATTENDRE
100 FOR J=0 TO 40
102 I=J/10
105 A=(360*I)/(2*(22/7)):A=INT(A+0.5)
110 B=SIN(I)
120 C=COS(I)
130 D=TAN(I)
135 PRINT USING "##.#";I;:PRINT TAB(6) USING "###.###";A;
136 PRINT TAB(14) USING "###.###";B;:PRINT TAB(21) USING "###.###";C;:
PRINT TAB(28) USING "###.###";D
137 IF J=0 THEN 150
140 IF J/10=INT(J/10) THEN GOSUB 2000
150 NEXT J
160 PRINT:PRINT"FIN":END
1000 REM UP EN-TETE
1010 PRINT:PRINT"  PI   DEGRE    SIN    COS    TAN":PRINT
1020 RETURN
2000 REM UP ATTENDRE
2010 LOCATE 5,23:PRINT"PRESSER UNE TOUCHE S'IL VOUS PLAIT!"
2020 A$=INKEY$:IF A$="" THEN 2020
2030 CLS
2040 GOSUB 1000
2050 RETURN
```

Liste des variables

A = angle (en degrés)
A$= variable chaîne de caractères de réponse
B = sinus de l'angle
C = cosinus de l'angle
D = tangente de l'angle
I = J/10 (valeur de l'abscisse des fonctions angulaires)
J = indice de boucle (ici de 0 à 40)

Description du programme

ligne 10-85	: titre
ligne 95	: appel du sous-programme d'attente (2000)
ligne 100-150	: calculs et sortie des résultats, 41 valeurs du tableau sont déterminées (J = 0 TO 40), les fonctions d'angle s'appliquent ici à I = J/10 (ligne 102)
ligne 105	: conversion de la valeur de l'angle (radians) en degrés
ligne 110-130	: détermination du sinus, cosinus et de la tangente
ligne 135-137	: sortie des résultats
ligne 140	: appel du sous-programme (2000) tous les 10 passages. Si J = 0 cet appel est évité (ligne 137), car il n'est pas encore utile. Dans ce cas il y a passage de la ligne 137 à la ligne 150
ligne 160	: fin du programme principal
ligne 1000-1020	: sous-programme de sortie de l'en-tête du tableau
ligne 2000-2050	: sous-programme d'attente (pour description voir tableau du programme 6, chapitre 1)

9.8 Tableau numérique 3

Enoncé du problème

Ce programme permet de sortir les logarithmes népériens ln(X) pour une variable X variant de 1 à 100.

Analyse du problème

La simplicité de l'énoncé nous dispense, ici aussi, d'analyser le problème.

Programme

```
10 REM Z7-TABLEAU 3
20 CLS
30 PRINT TAB(13)"TABLEAU NUMERIQUE 3":PRINT:PRINT
40 PRINT"PROGRAMME POUR DETERMINER LES VALEURS DES"
50 PRINT TAB(8)"LOGARYTHMES NEPERIENS"
60 PRINT:PRINT:PRINT TAB(10)"PROF.W.VOSS, 1984":PRINT
70 GOSUB 2000
80 FOR I=1 TO 100
85 L=LOG(I)
90 PRINT USING "####";I;:PRINT TAB(8) USING "###.####";L
100 IF I/15=INT(I/15) THEN GOSUB 2000
110 NEXT I
120 PRINT:PRINT:PRINT"FIN DE LA SORTIE":END
1000 REM UP EN-TETE
1010 PRINT:PRINT"NOMBRE LOGARYTHME ":PRINT
1020 RETURN
2000 REM UP ATTENDRE
2010 LOCATE 5,23:PRINT"PRESSER UNE TOUCHE S'IL VOUS PLAIT!"
2020 A$=INKEY$:IF A$="" THEN 2020
2030 CLS
2040 GOSUB 1000
2050 RETURN
```

Liste des variables

A$= variable chaîne de caractères de réponse
I = indice de boucle (ici 1 à 100)
L = logarithme népérien de I

Description du programme

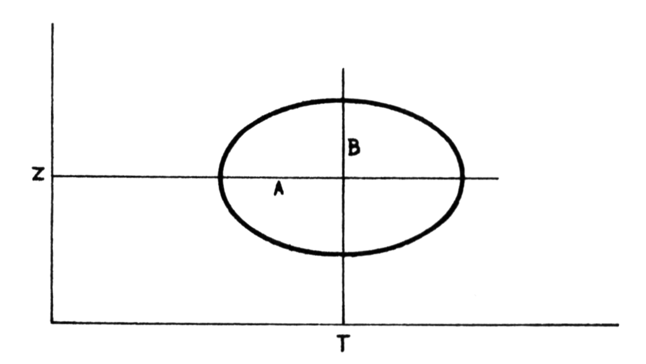
ligne 10-60 : titre
ligne 70 : appel du sous-programme d'attente (2000)
ligne 80-110 : calculs des logarithmes et sortie, appel du sous-programme d'attente (2000) toutes les 15 lignes du tableau
ligne 120 : fin du programme principal
ligne 1000-1020 : sous-programme pour la sortie de l'entête du tableau
ligne 2000-2050 : sous-programme d'attente

9.9 Ellipse

Enoncé du problème

Ce programme permet de dessiner n'importe quelle ellipse. Pour cela il nous faut entrer les valeurs suivantes :

T : colonne du centre de l'ellipse (0-319)
Z : ligne du centre de l'ellipse (0-199)
A : premier demi-axe de l'ellipse
B : deuxième demi-axe de l'ellipse



Analyse du problème

L'équation générale de l'ellipse, dans un repère passant par son centre est la suivante :

$$x^2 / a + y^2 / b = 1$$

Pour pouvoir représenter une fonction de ce genre à l'aide d'un programme BASIC, nous devons résoudre l'équation précédente d'après y.

D'où le résultat :

$$y_1 = b/a \sqrt{a^2 - x^2}$$

$$y_2 = - b/a \sqrt{a^2 - x^2}$$

Si nous prenons en compte le fait que le centre de l'ellipse a ici pour coordonnées (T,Z) (et non (0,0)), il est alors nécessaire d'effectuer ce même changement de repère à tout point de l'ellipse.

Programme

```
10 REM Z8-ELLIPSE
20 CLS
30 PRINT"PROGRAMME POUR DESSINER UNE"
40 PRINT TAB(16)"ELLIPSE QUELCONQUE"
50 PRINT:PRINT:PRINT TAB(9)"PROF,W,VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT"CE PROGRAMME REQUIERT COMME INFORMATIONS"
80 PRINT"INPUT:":PRINT
90 PRINT"COORDONEES DU CENTRE:":PRINT
100 PRINT TAB(5);:INPUT"VALEUR X (COLONNE):";T
110 PRINT TAB(5);:INPUT"VALEUR Y (LIGNE):";Z
120 PRINT:INPUT"PREMIER DEMI-AXE A:";A
125 PRINT:INPUT"DEUXIEME DEMI-AXE B:";B
130 CLS
140 PLOT 0,0:DRAW 639,0
150 PLOT 0,0:DRAW 0,399
160 PLOT 0,Z:DRAW 639,Z
170 PLOT T,0:DRAW T,399
390 REM ELLIPSE
400 C=T-A:D=T+A
410 IF C<0 THEN C=0
420 IF D>639 THEN D=639
430 FOR X=C+1 TO D-1
440 DS=(B/A)*SQR(A*A-(X-T)^2)
450 Y=Z+DS
460 IF Y<0 OR Y>399 THEN 480
470 PLOT X,Y
480 NEXT X
490 FOR X=D-1 TO C+1 STEP -1
500 DS=(B/A)+SQR(A*A-(X-T)^2)
510 Y=Z-DS
520 IF Y<0 OR Y>399 THEN 540
530 PLOT X,Y
540 NEXT X
550 LOCATE 5,23:PRINT"FIN":END
```

Liste des variables

A = premier demi-axe de l'ellipse
B = deuxième demi-axe de l'ellipse
C = valeur de l'extrémité gauche de l'ellipse
D = valeur de l'extrémité droite de l'ellipse
DS= grandeur auxiliaire résultante de l'équation de l'ellipse
T = colonne centre de l'ellipse
X = valeur de l'abscisse d'un point de l'ellipse
Y = valeur de l'ordonnée d'un point de l'ellipse
Z = ligne centre de l'ellipse

Description du programme

ligne 10-60	: titre
ligne 60-125	: entrées des coordonnées du centre de l'ellipse et des valeurs des demi-axes
ligne 130	: effacement de l'écran
ligne 140-150	: tracé du système de coordonnées
ligne 160-170	: tracé d'un deuxième système de coordonnées dont les axes passent par le centre de l'ellipse
ligne 390-480	: tracé de la moitié supérieure de l'ellipse
ligne 490-540	: tracé de la moitié inférieure de l'ellipse
ligne 550	: fin du programme

9.10 Répartition normalisée de Gauss

Enoncé du problème

Ce programme représente graphiquement la répartition normalisée de Gauss.

Sa fonction de densité est la suivante :

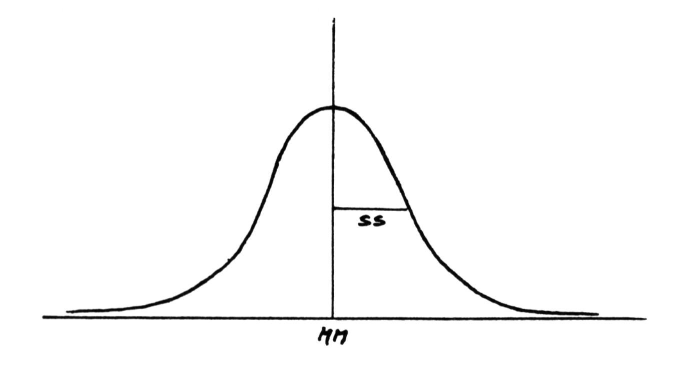
$$f(x) = \frac{1}{SS \cdot \sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{X - MM}{SS}\right)^2}$$

MM et SS sont les paramètres de position qui doivent être définis (voir le schéma suivant) :

MM désigne le centre de la répartition (courbe) et SS son étalement (écart standard = distance entre la verticale passant au centre et l'un des deux points d'inflexion de la répartition (courbe)).

Dans le programme suivant, nous avons pris :

MM = 200 et SS = 60



Programme

```
10 REM Z9-GAUSS
20 CLS
30 PRINT"PROGRAMME POUR LA REPRESENTATION D'UNE REPARTITION GAUSSIENNE
"
40 PRINT TAB(11)"NORMALISEE"
50 PRINT:PRINT:PRINT:PRINT TAB(9)"PROF.W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT"LE PROGRAMME DESSINE UN ECART NORMAL"
80 PRINT"AVEC LA VALEUR MOYENNE 200 ET"
90 PRINT"L'ECART STANDARD 60.":PRINT
100 PRINT"CES VALEURS PEUVENT ETRE MODIFIEES LE CAS ECHEANT"
110 PRINT TAB(10)"(VOIR LIGNE 120)"
115 GOSUB 1000:REM ATTENDRE
120 MM=200:SS=60
130 CLS
140 PLOT 0,0:DRAW 400,0
150 PLOT 200,0:DRAW 200,300
390 REM REPARTITION NORMALISEE
400 A=1/(SS*SQR(2*(22/7)))
410 FOR X=0 TO 400
420 B=((X-MM)/SS)^2
430 B=0.5*B*(-1)
440 Y=A*EXP(B)
450 Y=(2000000/SS)*Y
460 IF Y<0 OR Y>399 THEN 480
470 PLOT X,Y
480 NEXT
490 PRINT:END
1000 REM UP ATTENDRE
1010 LOCATE 5,23:PRINT"PRESSER UNE TOUCHE S'IL VOUS PLAIT!"
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS:RETURN
```

Liste des variables

A = premier terme de l'équation de la fonction
A$= variable chaîne de caractères de réponse
B = exposant de l'équation de la fonction
MM= valeur centre de la répartition normalisée
SS = écart standard de la répartition normalisée
X = abscisse d'un point de la courbe
Y = ordonnée d'un point de la courbe

Description du programme

ligne 10-110 : titre et explications
ligne 115 : appel du sous-programme d'attente
ligne 120-130 : présentation des paramètres de la fonction et effacement de l'écran
ligne 140-150 : tracé d'un système de coordonnées
ligne 390-480 : calcul des ordonnées de la répartition, normalisation (ligne 450) et sortie du graphique
ligne 490 : fin du programme
ligne 1000-1030 : sous-programme d'attente

9.11 Régression

Enoncé du problème

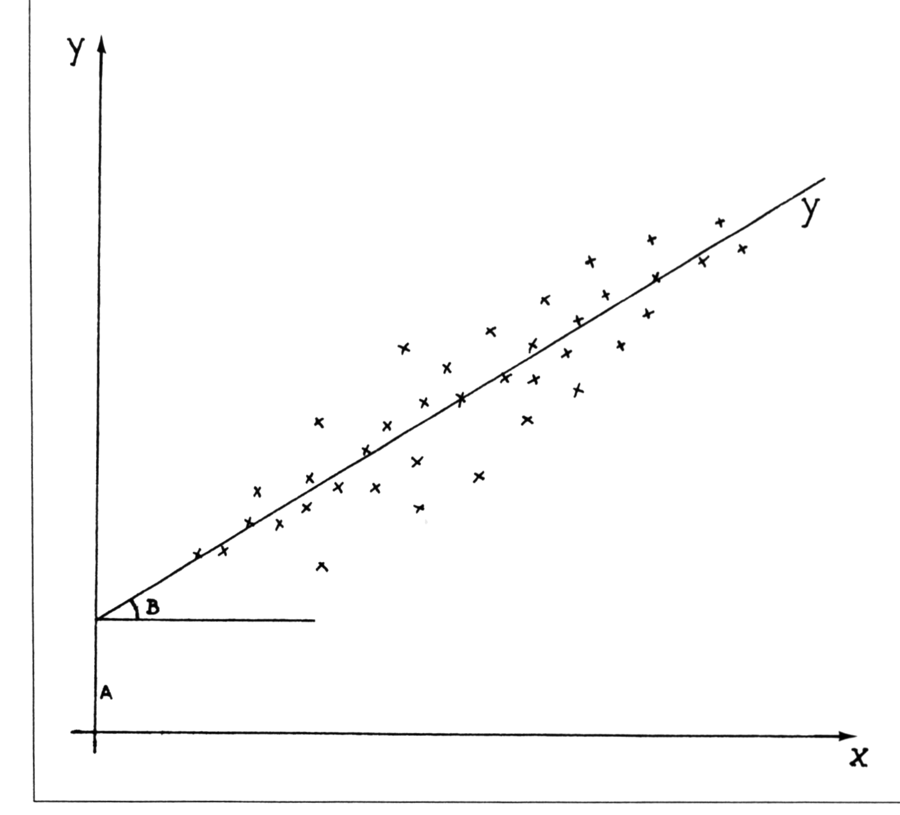
La régression a une grande importance dans le domaine des statistiques mathématiques. Il s'agit de décrire une répartition statistique bivariante à l'aide d'une fonction mathématique simplifiée, une droite dans le cas le plus simple.

Cette répartition provient par exemple d'une enquête auprès d'une population donnée, celle-ci portant simultanément sur deux points (variables).

Exemple : Demande de la taille et du poids des personnes interrogées.

Le but d'un tel sondage est de voir s'il existe un lien entre ces deux variables.

Il est possible de représenter cette répartition bivariante par un "nuage de points" dans un système de coordonnées à deux dimensions, semblable à celui du schéma suivant :



Pour décrire la relation existant entre ces deux variables, on essaie de trouver une fonction mathématique linéaire telle que les points du nuage soient les plus proches et les mieux répartis possible autour de la représentation de cette fonction.

A ce sujet, il existe également des formules arithmétiques non-linéaires (régression multiple) que nous n'abordons pas ici.

Analyse du problème

On détermine la fonction, linéaire ou non, à l'aide de la méthode des moindres carrés.

Cette méthode offre une formule de calcul qui permet de déterminer les paramètres de la fonction à adapter. Dans le cas d'une droite il s'agit du segment des ordonnées A et de l'angle B (voir schéma précédent).

La méthode des moindres carrés permet de déterminer ces paramètres tels que la somme des carrés des écarts à la moyenne soit minimum. D'ou la formule :

$$\sum(y_i - y_i')^2 = \text{min.}$$

Comme les valeurs Yi' se calculent d'après l'équation de la droite :

$$y_i' = a + b * x_i$$

et d'après la règle de minimisation, il s'ensuit :

$$\sum(y_i - a - b*x_i)^2 = \text{min.}$$

Pour trouver le minimum de cette relation on calcule ses dérivées partielles d'après a et b (la dérivée de " Min." (constante) est bien sûr nulle). Puis l'on résoud les deux équations à deux inconnues (a et b) ainsi obtenues.

D'ou :

$$b = \frac{n * \sum x_i y_i - \sum x_i \sum y_i}{n * \sum xi^2 - (\sum x_i)^2}$$

$$a = \frac{\sum y_i}{n} - b * \frac{\sum x_i}{n}$$

Le programme BASIC qui est développé permet de calculer ces deux paramètres pour toute paire de valeurs (x_i,y_i) à entrer, ceci d'après la méthode des moindres carrés. Ce sont les valeurs de ces deux paramètres qui sont données en résultat.

Programme

```
10 REM Z10-REGRESSION
20 CLS
30 PRINT"PROGRAMME DETERMINANT LES PARAMETRES D'UNE FONCTION"
40 PRINT"LINEAIRE DE REGRESSION OU D'UNE LIGNE A TENDANCE"
50 PRINT TAB(6)"LINEAIRE"
60 PRINT:PRINT:PRINT TAB(9)"PROF.  W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT:PRINT
80 PRINT"CE PROGRAMMME REQUIERT DES PAIRES DE VALEURS EN TANT"
90 PRINT"QU'INFORMATION D'ENTREE (PAR EXEMPLE VALEURS X ET Y"
100 PRINT"OU VALEURS DE TEMPS ET D'INDICE.ETC..)"
110 GOSUB 1000:REM ATTENDRE
120 INPUT"COMBIEN DE PAIRES DE VALEURS:";N
130 DIM X(N),Y(N)
140 PRINT:PRINT TAB(5)"ENTREE : ":PRINT:PRINT
150 FOR I=1 TO N
160 PRINT I;".PAIRE X,Y : ";:INPUT X(I),Y(I)
170 NEXT I
172 CLS:PRINT"CONTROLES ":PRINT:PRINT
174 PRINT"NO:         VALEURS X   VALEURS Y":PRINT
176 FOR I=1 TO N:PRINT I,X(I),Y(I)
178 IF I/10=INT(I/10) THEN GOSUB 1000
179 NEXT I
180 REM REGRESSION
185 GOSUB 1000:REM ATTENDRE
190 FOR I=1 TO N
200 S1=S1+X(I)*Y(I)
210 S2=S2+X(I):S3=S3+Y(I)
220 S4=S4+X(I)*X(I)
230 NEXT I
240 B=(N*S1-S2*S3)/(N*S4-S2*S2)
250 A=S3/N-B*S2/N
265 CLS
270 PRINT"RESULTAT : ":PRINT:PRINT
280 PRINT"SEGMENT DES ORDONNEES A = ";A
290 PRINT:PRINT"                          B = ";B
```

```
300 PRINT:PRINT:PRINT:PRINT
310 PRINT"CES PARAMETRES PERMETTENT D'APPELER LE"
320 PRINT"PROGRAMME POUR DESSINER UNE"
330 PRINT"DROITE,SI L'ON SOUHAITE ILLUSTRER "
340 PRINT"LE RESULTAT PAR UN GRAPHIQUE."
350 PRINT:PRINT:PRINT"FIN DE LA SORTIE":END
1000 REM ATTENDRE
1010 LOCATE 5,23:PRINT"PRESSER ENCORE UNE TOUCHE!"
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
```

Liste des variables

A = segment des ordonnées de la droite
A$= variable chaîne de caractères de réponse
B = angle que forme la droite avec l'horizontale passant par son point d'origine
I = indice de boucle
N = nombre de paires de valeurs
S1 = sommes auxiliaires pour les calculs de B et A
S2 = " "
S3 = " "
S4 = " "
X = tableau à une dimension des abscisses des points
Y = tableau à une dimension des ordonnées des points

Description du programme

ligne 10-110	: titre, explications et appel du sous programme d'attente
ligne 120-130	: entrée du nombre de paires de valeurs indicées et dimensionnement des tableaux X et Y
ligne 140-170	: entrée des valeurs indicées (INPUT)
ligne 172-179	: sortie de contrôle des paires de valeurs indicées et pause toutes les dix paires
ligne 180-185	: appel du SP d'attente
ligne 190-230	: calcul des quatre sommes qui sont utilisées pour déterminer A et B
ligne 240-250	: calcul des paramètres A et B de la droite
ligne 265-300	: affichage du résulat
ligne 310-350	: message et fin du programme
ligne 1000-1040	: sous-programme d'attente

9.12 Permutations

Enoncé du problème

Le calcul des permutations permet de déterminer le nombre de façons possibles pour classer des éléments différents.

Analyse du problème

Prenons par exemple trois éléments différents les uns des autres a, b et c. Leurs dispositions possibles sont les suivantes :

abc, acb, bac, bca, cab, cba

Pour trois éléments il existe donc six façons différentes de les organiser.

Le raisonnement général est le suivant :

Pour le premier des trois éléments il existe trois possibilités, il peut se situer en première, seconde ou troisième position.

Pour le second élément, après avoir décidé de la place du premier, il n'en reste déjà plus que deux.

Ceci signifie qu'il existe au total pour les deux premiers éléments 3 * 2 = 6 possibilités.

Il n'en reste plus qu'une pour le troisième élément, ce qui donne donc 3 * 2 * 1 = 6 possibilités de disposition différentes.

Si maintenant on a n éléments, nous avons donc un nombre de possibilités de disposition égal à :

n * (n - 1) * (n - 2) * (n - 3) * ... * 3 * 2 * 1

Ceci se note n ! (factoriel n) et l'on dit que n éléments différents peuvent être disposés de n ! façons différentes ; Ou encore que la permutation de n éléments différents correspond à n ! possibilités.

Attention : Pour les très grandes valeurs de n, l'expression n ! augmente extrêmement rapidement, si vite que l'ordinateur ne peut plus en représenter le résultat (à partir de n = 34) !

Programme

```
10 REM Z11-PERMUTATIONS
20 CLS
30 PRINT"PROGRAMME DETERMINANT LE NOMBRE DE"
40 PRINT"PERMUTATIONS DE N ELEMENTS."
50 PRINT TAB(5)"DIFFERENTS(=FACTORIELLE N = N!)"
60 PRINT:PRINT:PRINT TAB(9)"PROF.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT
80 INPUT"NOMBRE DES ELEMENTS N : ";N
90 P=N
100 IF N>=34 THEN PRINT"N EST TROP GRAND POUR LES CAPACITES DE CET ORD
INATEUR":GOTO 150
110 FOR I=N-1 TO 1 STEP -1
120 P=P*I
130 NEXT I
140 PRINT:PRINT"FACTORIELLE DE ";N;" EST ";P
150 PRINT:PRINT:PRINT"ENCORE UN CALCUL?(O/N)";:INPUT A$
160 IF A$="O" THEN CLS:GOTO 80
170 PRINT:PRINT:PRINT"FIN":END
```

Liste des variables

A$= variables chaine de caractères de réponse

I = indice de boucle (termes du produit)

N = nombre d'éléments

P = variable de calcul intermédiaire et résultat final

Description du programme

ligne 10-70 : titre et explications
ligne 80-90 : demande d'entrée du nombre d'éléments N à permuter. Initialisation à N de la variable P (premier terme de la formule).
ligne 100 : message (si N est supérieur ou égal à 34, les capacités de l'ordinateur sont trop faibles) saut à la ligne 150
ligne 110-130 : calcul du factoriel (dans P)
ligne 140 : sortie du résultat
ligne 150-170 : fin du programme à moins qu'un nouveau calcul ne soit souhaité (dans ce cas retour à la ligne 80 après effacement de l'écran)

9.13 Combinaisons

Enoncé du problème

Le problème abordé ici est similaire au précédent :
il s'agit de calculer de combien de façons il est possible de sélectionner k éléments à partir de n éléments différents.

La formule de calcul est la suivante :

Nombre = n ! / k ! * (n-k) !

On appelle cette valeur "coefficient binômial" et on la note :

$$\binom{n}{k}$$

Elle correspond d'ailleurs aux termes qui proviennent de la résolution des polynômes.

Exemples :

$$(a + b)^2 = a^2 + 2ab + b^2$$

$$= 1a^2b^0 + 2a^1b^1 + 1a^0b^2$$

$$= \binom{2}{0}a^2b^0 + \binom{2}{1}a^1b^1 + \binom{2}{2}a^0b^2$$

$$(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

$$= 1a^3b^0 + 3a^2b^1 + 3a^1b^2 + 1a^0b^3$$

$$= \binom{3}{0}a^3b^0 + \binom{3}{1}a^2b^1 + \binom{3}{2}a^1b^2 + \binom{3}{3}a^0b^3$$

Tout binôme peut s'écrire sous la forme d'une somme de termes. Le nombre de termes de cette somme est obtenu en ajoutant un à la valeur de la puissance du binôme (dans l'exemple (2) la puissance est 3 et la somme comprend 4 termes).

Chaque terme se compose de 3 facteurs :

- une constante (le coefficient binômial),
- un terme "a" élevé à une puissance qui décroit de la puissance du polynôme (ici 3) à zéro.
- un terme "b" élevé à une puissance qui croit de zéro à la puissance du polynome (ici 3).

Dans le coefficient binômial (n k), n est égal au degré du polynome (3) et k est un entier variant de zéro à n (voir l'exemple précédent).

Le programme calcule des coefficients binômiaux quelconques. On peut donc s'en servir pour l'écriture de binômes de faible degré ou de degré important (ex : $(a+b)^{27}$).

Mais ce programme peut également être employé pour en effectuer d'autres (voir les deux sections précédentes).

Enfin un programme de ce genre permet par exemple de répondre à la question :

"8 coureurs commencent un 100 mètres. De combien de manières les trois premières places peuvent-elles être occupées ?"

Réponse :

$$\binom{8}{3} = 8\,! / (3\,! * (8-3)\,!) = 8\,! / (3\,! * 5\,!)$$

$$= (8 * 7 * 6) / (3 * 2 * 1) = 56$$

Programme

```
10 REM Z12-COMBINAISONS
20 CLS
30 PRINT"PROGRAMME DETERMINANT LE NOMBRE DE"
40 PRINT"COMBINAISONS DE K A PARTIR DE N ELEMENTS."
50 PRINT:PRINT:PRINT"ON PARLE A CE SUJET DU"
55 PRINT"COEFFICIENT BINOMIAL"
60 PRINT:PRINT:PRINT:PRINT TAB(9)"PROF.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT
80 INPUT"NOMBRE D'ELEMENTS N: ";N
90 INPUT"NOMBRE D'ELEMENTS A SELECTIONNER K: ";K
100 Y=N-K
110 IF K=0 OR K=N THEN BK=1:GOTO 210
120 IF K>N/2 THEN H=Y:Y=K:K=H
130 GOSUB 500
140 PX=A:A=N
150 IF PX=1 THEN PN=N:GOTO 200
160 FOR I=N-1 TO N-K+1 STEP -1
170 A=A*I
180 NEXT I
190 PN=A
200 BK=PN/PX
210 PRINT:PRINT:PRINT"NOMBRE DES COMBINAISONS DE";K;"A PARTIR DE"
220 PRINT N;"ELEMENTS EST: ";BK
230 PRINT:PRINT:PRINT"FIN DES CALCULS":END
500 REM UP FAK
510 A=K
520 IF K=0 OR K=1 THEN A=1:GOTO 560
530 FOR I=K-1 TO 1 STEP -1
540 A=A*I
550 NEXT I
560 RETURN
```

Liste des variables

A = variable pour le calcul factoriel
(produit intermédiaire et final)
BK= coefficient binômial
H = valeur auxiliaire pour la permutation de K et de Y
I = indice de boucle (facteur du calcul factoriel)
K = nombre d'éléments à sélectionner
N = nombre d'éléments
PN= premier terme pour le calcul du coefficient binômial
(numérateur cf. formule)
PX= deuxième terme pour le calcul du coefficient binômial
(dénominateur cf. formule)
Y = N - K

Description du programme

ligne 10-70 : titre et explications
ligne 80-90 : entrée (nombre d'éléments N et nombre K des éléments à sélectionner)
ligne 100 : initialisation de la variable auxiliaire Y
ligne 110 : initialisation de la variable BK à un, pour K = 0 ou K = N, étant donnée la définition suivante :

$$\binom{N}{0}=\binom{N}{N}=1$$

ligne 120 : permutation de K et (N-K) (=Y), si K>N/2. Ceci est possible parce que :

$$\binom{N}{K} = N\,!\,/\,(K\,!\,*\,(N-K)\,!) = \binom{N}{N-K}$$

ligne 130 : appel du SP 500 pour le calcul de K !
ligne 140-190 : calcul du second terme binômial après la simplification appropriée (voir l'exemple des coureurs)
ligne 200-230 : calcul et sortie du coefficient binômial et fin du programme principal
ligne 500-560 : sous-programme de calcul de K !

9.14 Loto

Enoncé du problème

Le programme qui est élaboré ici permet d'évaluer nos chances de gagner au jeu du loto.

Analyse du problème

Ce programme calcule cette probabilité de gagner pour 3, 4, 5 ou 6 bons numéros au loto (sur 49).

Elle est obtenue à l'aide de la formule suivante :

$$P(x) = \frac{\binom{6}{x}\binom{43}{6-x}}{\binom{49}{6}} = (B1 * B2) / B3$$

où X indique "le nombre de bons numéros" donné par l'utilisateur.

Nous travaillons donc ici avec trois coefficients binômiaux (B1, B2, B3) déjà vus dans le chapitre précédent.

Programme

```
10 REM Z13-LOTO
20 CLS
30 PRINT"PROGRAMME DONNANT LA PROBABILITE POUR 3,4,5"
40 PRINT"OU 6 BONS NUMEROS AU "
50 PRINT"LOTO  (6/49)."
60 PRINT:PRINT:PRINT:PRINT TAB(10)"PROF.W.VOSS 1984"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"CE PROGRAMME NE TIENT PAS COMPTE DE LA POSSIBILITE"
90 PRINT"DE 5 BON NUMEROS AVEC LE NUMERO COMPLEMENTAIRE"
100 PRINT
110 GOSUB 1000:REM ATTENDRE
120 INPUT "COMBIEN DE NUMEROS";X
125 IF X=6 THEN B1=1:B2=1:GOTO 300
130 REM B1
135 IF X=5 THEN B1=6:B2=43:GOTO 300
140 Y=6-X
150 P=X:FOR I=X-1 TO 1 STEP -1
160 P=P*I:NEXT I:P1=P
170 P=Y:FOR I=Y-1 TO 1 STEP -1
180 P=P*I:NEXT I
190 P2=P
200 B1=720/(P1*P2)
210 REM B2
220 X=Y
230 P=43
240 FOR I=P-1 TO 43-X+1 STEP-1
250 P=P*I:NEXT I
260 P1=P
265 P=X
270 FOR I=X-1 TO 1 STEP -1
280 P=P*I:NEXT I
290 P2=P:B2=P1/P2
300 REM B3
310 P1=49:P2=6
320 FOR I=1 TO 5
```

```
330 P1=P1*(49-I)
340 P2=P2*(6-1)
350 NEXT I
360 B3=P1/P2:W=(B1*B2)/B3
370 CLS
380 PRINT"     RESULTATS  : ":PRINT:PRINT
390 PRINT"LA PROBABILITE DE SORTIE DE";X;"BONS NUMEROS"
400 PRINT"AU LOTO S'ELEVE A       : "
410 PRINT:PRINT:PRINT TAB(5)W
420 PRINT:PRINT:PRINT"ENCORE UN CALCUL  (O/N)  ";:INPUT A$
430 IF A$="O" THEN CLS: GOTO 120
440 PRINT:PRINT:PRINT"FIN":END
1000 REM ATTENDRE
1010 LOCATE 5,23:PRINT"PRESSEZ UNE TOUCHE  S.V.P"
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
```

Liste des variables

A$= variable chaîne de caractères de réponse
B1 = premier terme de la formule arithmétique
B2 = deuxième terme de la formule arithmétique
B3 = troisième terme de la formule arithmétique
P = variable pour les calculs de factoriels
(produit intermédiaire)
P1 = premier terme pour le calcul d'un coefficient binômial
P2 = deuxième terme pour le calcul d'un coefficient binômial
W = probabilité (résultat final)
X = nombre de "bons numéros"
Y = 6 - X

Description du programme

ligne 10-100	: titre et explications
ligne 110	: appel du sous-programme d'attente (1000)
ligne 120	: entrée du nombre de bons numéros
ligne 125	: si X = 6 alors B1 = B2 = 1 (voir programme Z10) poursuite à la ligne 300
ligne 130-200	: détermination du coefficient binomial B1 (voir programme Z10)
ligne 210-290	: idem pour B2
ligne 300-360	: idem pour B3
ligne 360	: calcul de la probabilité W
ligne 370-410	: affichage du résultat
ligne 420-440	: fin du programme principal à moins qu'un autre calcul ne soit souhaité
ligne 1000-1040	: sous-programme d'attente

9.15 Répartition binômiale

Enoncé du problème

La répartition binômiale est une répartition des probabilités. Elle sert de fondement à tout un éventail de procédures de calcul des probabilités.

Elle permet de répondre à la question suivante :

Quelle est la probabilité pour que sur n essais indépendants un évènement favorable se produise x fois sachant que deux solutions sont possibles pour chaque tentative et qu'une solution ne se réalise que sur P % d'essais ?

Prenons un exemple simple :

En lançant 10 fois une pièce de monnaie, combien de chances existent pour qu'elle retombe 5 fois du côté face ?

Ceci est un cas d'application de la répartition binômiale puisque pour chaque essai il y a deux possibilités ("pile" ou "face") et que les essais (quand on lance la pièce) sont indépendants les uns des autres.

Autre exemple :

Quelle est la probabilité pour que, sur 5 naissances, 2 filles et 3 garçons viennent au monde ?

Analyse du problème

Pour analyser le problème posé, prenons le dernier exemple mentionné :

FFGGG (F = fille, G = garçon), GGGFF sont des représentations possibles des combinaisons "d'enfants" répondant à la question posée. Il y en a beaucoup d'autres.

Le nombre total de combinaisons possibles peut être trouvé à l'aide du calcul de permutation dont nous avons déjà parlé.

Sans vouloir le démontrer, nous constatons que le nombre des différentes combinaisons possibles à partir de deux filles et de trois garçons s'obtient de la manière suivante :

nombre = 5 ! / (2 ! * 3 !)

= 5 * 4 * 3 * 2 * 1 / (2 * 1 * 3 * 2 * 1)

= 5 * 4 / (2 * 1) = 10

Il est possible de dire d'une manière générale que le nombre de dispositions possibles de x_i éléments identiques pris parmi n éléments avec ($n-x_i$) éléments identiques, différents des x_i autres, peut s'obtenir d'après la formule suivante :

nombre = n ! / (x_i ! * ($n-x_i$) !)

Pour continuer l'analyse du problème, nous considérons que la probabilité de naissance d'une fille est de P (P = 0.5).

Examinons encore une fois ce résultat :

FFGGG

Combien de chances y-a-t'il pour qu'un tel évènement arrive ?

Envisageons d'abord la naissance d'une première fille : sa probabilité est bien sûr égal à 0.5.

Celle de la naissance d'une seconde fille s'élève aussi à 0.5.

Maintenant la probabilité pour que les deux premières naissances soient celles de filles est alors de 0.5 * 0.5 (de manière générale P * P).

La probabilité pour qu'un garçon naisse s'élève à 1 - 0.5 (1 - P). En raisonnant de la même manière la probabilité d'obtenir un garçon à la troisième, quatrième et cinquième naissance est :

(1 - 0.5) * (1 - 0.5) * (1 - 0.5)

ou plus généralement

(1 - P) * (1 - P) * (1 - P)

Pour la combinaison FFGGG nous obtenons alors au total la probabilité suivante notée d'une manière générale comme ceci :

$$P * P * (1 - P) * (1 - P) * (1 - P)$$

$$= P^2 * (1 - p)^3$$

$$= P^{xi} * (1 - P)^{n - xi}$$

Le résultat de notre exemple est donc :

probabilité d'avoir FFGGG =

$$0.5^2 * (1 - 0.5)^{5 - 2} = 0.5^5 = 1/32$$

Cependant nous avons vu qu'il y a 10 combinaisons possibles répondant au problème.

C'est à partir de ce nombre que l'on trouve la probabilité définitive, puisque pour chacune des 10 combinaisons on obtient la probabilité de 1/32, ainsi :

Probabilité d'avoir 2 filles sur cinq naissances est égale à :

10 * 1/32 = 10 /32 = 31.25 %

Nous pouvons déduire maintenant de cet exemple la formule arithmétique générale :

La probabilité d'avoir x_i évènements favorables pour un nombre n d'essais, sachant que la probabilité par essai, pour qu'un évènement favorable arrive, est de P, s'obtient par cette formule :

$$W = (n ! / (x_i ! * (n-x_i) !)) * P^{xi} * (1-P)^{n-xi}$$

Malgré sa complexité, cette expression est facile à programmer.

Programme

```
10 REM Z14-BINOMINALE
20 CLS
30 PRINT"PROGRAMME POUR LE CALCUL DES PROBABILITES"
40 PRINT"A L'AIDE DE LA REPARTITION"
50 PRINT"BINOMINALE"
60 PRINT:PRINT:PRINT TAB(10) "PROF.W.VOSS 1984"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"CE PROGRAMME OU PLUTOT LA"
90 PRINT"LA REPARTITION BINOMINALE PERMET DE"
100 PRINT"REPONDRE A LA QUESTION SUIVANTE :":PRINT
110 PRINT"QUELLE EST LA PROBABILITE POUR QU'AVEC"
120 PRINT"'N' ESSAIS, L'EVENEMENT APPROPRIE SE MANIFESTE"
130 PRINT"'X' FOIS QUAND LA PROBABILITE D'UN"
140 PRINT"ESSAI S'ELEVE A 'P'"
150 PRINT
160 GOSUB 1000:REM ATTENDRE
170 PRINT"CE PROGRAMME REQUIERT COMME"
180 PRINT"INFORMATION  :":PRINT
190 PRINT:INPUT "NOMBRE D'ESSAIS                : ";N
200 PRINT:INPUT "NOMBRE D'EVENEMENTS APPROPRIES : ";X
220 PRINT:PRINT"PROBABILITE DE MANIFESTATION POUR"
230 INPUT "UN ESSAI                       : ";P
240 REM COEFFICIENT BINOMINAL
245 Y=N-X:Q=1-P
250 IF X=0 OR X=N THEN BK=1:GOTO 360
260 IF X > N/2 THEN H=Y:Y=X:X=H:G=Q:Q=P:P=G
270 F=X:GOSUB 500:REM FACTORIELLE
290 PX=A:A=N
295 IF PX=1 THEN PN=N:GOTO 350
300 FOR I=N-1 TO N-X+1 STEP -1
310 A=A*I
320 NEXT I
330 PN=A
350 BK=PN/PX
360 W=BK*P^X*Q^Y
```

```
370 PRINT"   RESULTAT   :":PRINT:PRINT
380 PRINT"LA PROBABILITE QUE POUR              : ";N
390 PRINT"ESSAIS, L'EVENEMENT APPROPRIE SE MANIFESTE"
400 PRINT X;"FOIS S'ELEVE A                    : "
410 PRINT:PRINT:PRINT TAB (5) W
420 PRINT:PRINT:PRINT"ENCORE UN CALCUL  (O/N) ";:INPUT A$
430 IF A$="O" THEN CLS:GOTO 190
440 PRINT:PRINT:PRINT"FIN":END
500 REM FACTORIELLE
510 A=X
520 IF X=0 OR X=1 THEN A=1:GOTO 560
530 FOR I=X-1 TO 1 STEP -1
540 A=A*I
550 NEXT I
560 RETURN
1000 REM ATTENDRE
1010 LOCATE 5,23:PRINT"PRESSEZ UNE TOUCHE  S.V.P"
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
```

Liste des variables

A = variable pour le calcul d'une factoriel
A$= variable chaîne de caractères de réponse
BK= coefficient binômial
F = valeur auxiliaire (F - X)
G = valeur auxiliaire de travail pour la permutation de P avec Q
H = valeur auxiliaire de travail pour la permutation de X avec Y
I = indice de boucle
N = nombre d'essais
P = probabilité de manifestation d'un évènement favorable par essai
PN= premier terme du coefficient binômial
PX= deuxième terme du coefficient binômial
Q = 1 - P
W = probabilité finale (résultat)
X = nombre d'évènements favorables à obtenir
Y = N - X

Description du programme

ligne 10-150 : titre et explications
ligne 160-230 : attente puis entrée des informations (N, X, P)
ligne 240-350 : détermination du coefficient binômial
ligne 245 : initialisation de Y et Q
ligne 250 : si X = 0 ou X = N, alors BK est égal à 1, poursuite en 360
ligne 260 : si X > N/2, alors transposition des variables X avec N-X et P avec 1-P
ligne 270 : détermination de X !
ligne 290 : initialisation de PX (dénominateur de BK) à A puis de A à N

ligne 295 : si le dénominateur de BK = 1, alors le numérateur de BK est initialisé à N ; poursuite à la ligne 350
ligne 300-320 : détermination du numérateur du coefficient binômial BK
ligne 350 : calcul de BK
ligne 360 : calcul de la probabilité recherchée
ligne 365-410 : affichage du résultat
ligne 420-499 : fin du programme, à moins qu'un nouveau calcul ne soit souhaité (alors, effacement de l'écran et retour à la ligne 190)
ligne 500-560 : sous-programme de calcul d'une factoriel
ligne 1000-1040 : sous-programme d'attente

9.16 Elections

Enoncé du problème

Ce programme représente concrètement les résultats d'élections par un diagramme construit avec des barres.

Pour ce faire, l'utilisateur doit entrer les pourcentages de voix obtenus par les différents partis : les jaunes, les rouges, les bleus, les verts et les autres.

Analyse du problème

Une fois encore une analyse du problème serait inutile, étant donnée la simplicité de la tâche.

Programme

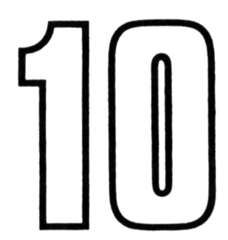
```
10 REM Z16-ELECTIONS
20 CLS
30 PRINT"PROGRAMME REPRESENTANT PAR UN GRAPHIQUE"
40 PRINT TAB(9)"LES RESULTATS ELECTORAUX"
50 PRINT:PRINT:PRINT
60 PRINT TAB(9)"PROF.  W.VOSS, 1984"
70 PRINT:PRINT:PRINT
80 PRINT"IL FAUT ENTRER LES POURCENTAGES DES RESULTATS DES ELECTIONS."
90 PRINT:PRINT:PRINT
95 DIM F(5),G(5)
100 INPUT"JAUNES : ";F(1)
110 INPUT"ROUGES : ";F(2)
120 INPUT"BLEUS  : ";F(3)
130 INPUT"VERTS  : ";F(4)
140 INPUT"AUTRES : ";F(5)
150 CLS
160 PLOT 0,110:DRAW 600,110
180 FM=0:FOR I=1 TO 5:IF F(I)>FM THEN FM=F(I)
190 NEXT I
200 FOR I=1 TO 5:G(I)=(17/FM)*F(I)
210 G(I)=INT(G(I)+0.5):NEXT I
230 GOSUB 2000:REM DESSIN
250 LOCATE 1,22
260 PRINT"JAU.  ROU.  BLE.  VER.  AUTRES"
270 FOR J=1 TO 5:PRINT TAB(6*(J-1))F(J);:NEXT J
280 PRINT:PRINT"FIN":END
2000 REM DESSIN
2010 FOR J=1 TO 5
2020 B=17:A=18-G(J)
2030 FOR I= A TO B
2040 LOCATE (J-1)*6+2,I:PRINT CHR$(143)
2050 LOCATE (J-1)*6+3,I:PRINT CHR$(143)
2060 NEXT I
2070 NEXT J
2080 RETURN
```

Liste des variables

A = ligne du début du dessin
B = ligne de la fin du dessin
F = tableau de pourcentages
FM= valeur maximale
G = tableau des pourcentages normalisés
I = indice de boucle
J = indice de boucle

Description du programme

ligne 10-90 : titre et explications
ligne 95 : dimensionnement des tableaux F et G
ligne 100-140 : entrée des informations (pourcentage de voix obtenu par les différents "partis")
ligne 150-160 : effacement de l'écran et tracé d'une ligne horizontale
ligne 180 : recherche du plus grand pourcentage parmi ceux qui ont été entrés par l'utilisateur
ligne 200-210 : calcul des pourcentages de manière à ce que le plus grand soit représenté par la plus grande colonne visualisable sur l'écran
ligne 230 : saut dans le SP 2000 de visualisation graphique (tracé des barres)
ligne 250-270 : sortie de la liste des différents "partis" et des pourcentages au bas du dessin
ligne 280 : fin du programme principal
ligne 2000-2090 : sous-programme pour le tracé des barres
ligne 2010-2020 : détermination de l'adresse initiale et finale de la barre
ligne 2030-2070 : représentation inversée d'espaces aux endroits déterminés par LOCATE
ligne 2090 : fin du sous-programme, retour en arrière



PERSPECTIVES

Nous avons tenté de montrer au cours des chapitres précédents les différentes façons de résoudre, à partir de programmes BASIC relativement simples, des problèmes types, semblables à ceux qui sont posés à l'école.

Il est nécessaire d'insister sur le fait que les exemples choisis restent des exemples et, qu'il n'était pas de notre propos de faire des programmes performants.

Il s'agissait plutôt de permettre au lecteur de comprendre rapidement comment fonctionne un programme. Nous espérons que cela l'aidera à trouver la solution à d'autres problèmes souvent semblables à ceux qui ont été présentés ici à titre d'exemple.

En fait, dans l'utilisation d'un ordinateur le plus important n'est pas l'ordinateur lui-même mais l'analyse des problèmes par l'utilisateur afin de les lui soumettre, sous forme de programme, pour lui permettre de les résoudre.

Ce qui est primordial n'est pas qu'un programme se déroule sans problème dès le premier essai (bien que cela soit très encourageant), mais que l'utilisateur sache bien construire ses propres programmes à partir des problèmes posés.

Un ordinateur ne sert vraiment à rien si vous n'avez pas déjà en tête la manière dont vous allez vous y prendre pour résoudre le problème.

Et si un jour un programme devait ne pas fonctionner comme vous vous y attendiez et que son déroulement est interrompu par un message "Système" d'erreur par exemple, il est toujours temps, alors, de rejeter un oeil sur le manuel d'utilisation.

Nous aimerions attirer l'attention des lecteurs qui ont trouvé trop simples nos exemples, sur le fait que ce livre a été conçu avant tout à l'intention des gens qui débutent en BASIC et que nous ne voulions pas décourager par la complexité éventuelle de nos énoncés. Ils se consoleront en apprenant qu'un second tome va paraître dans lequel figureront d'autres problèmes qui seront nettement plus difficiles et axés principalement sur les mathématiques.

INDEX

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

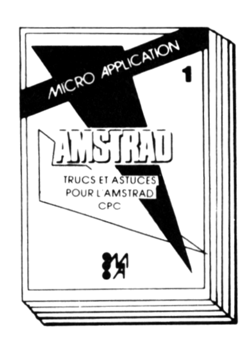
S

T

V

Z

les livres Amstrad



TRUCS ET ASTUCES POUR L'AMSTRAD CPC (Tome 1)

C'est le livre que tout utilisateur d'un CPC doit posséder. De nombreux domaines sont couverts (graphismes, fenêtres, langage machine) et des super programmes sont inclus dans ce best-seller (gestion de fichiers, éditeur de textes et de sons...).
Réf. : ML 112
Prix : **149** FF



PROGRAMMES BASIC POUR LES CPC (Tome 2)

Alimentez votre CPC
Ce livre contient de supers programmes, notamment un désassembleur, un éditeur graphique, un éditeur de texte... Tous les programmes sont prêts à être tapés et abondamment commentés.
Réf. : ML 118
Prix : **129** FF

LE BASIC AU BOUT DES DOIGTS CPC (Tome 3)

Ce livre est une introduction complète et didactique au BASIC du micro-ordinateur AMSTRAD CPC 464. Il permet d'apprendre rapidement et facilement la programmation (instructions BASIC, analyses des problèmes, algorithmes complexes...).
Comprenant de nombreux exemples, ce livre vous assure un apprentissage simple et efficace du BASIC CPC.
Réf. : ML 119
Prix : **149** FF



AMSTRAD OUVRE-TOI (Tome 4)

Le bon départ avec le CPC 464 ! Ce livre vous apporte les principales informations sur l'utilisation, les possibilités de connexions du CPC 464 et les rudiments nécessaires pour développer vos propres programmes. C'est le livre idéal de tous ceux qui veulent pénétrer dans l'univers des micro-ordinateurs avec le CPC
Réf. : ML 120
Prix : **99** FF





JEUX D'AVENTURES. COMMENT LES PROGRAMMER (Tome 5)

Voici la clé du monde de l'aventure. Ce livre fournit un système d'aventures complet, avec éditeur, interpréteur, routines utilitaires et fichiers de jeux. Ainsi qu'un générateur d'aventures pour programmer vous-mêmes facilement vos jeux d'aventures. Avec, bien sûr, des programmes tout prêts à être tapés.
Réf. : ML 121
Prix : **129** FF



LA BIBLE DU PROGRAMMEUR DE L'AMSTRAD CPC (Tome 6)

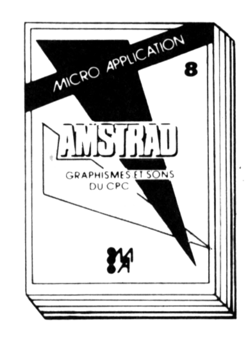
Tout, absolument tout sur le CPC 464. Ce livre est l'ouvrage de référence pour tous ceux qui veulent programmer en pro leur CPC. Organisation de la mémoire, le contrôleur vidéo, les interfaces, l'interpréteur et toute la ROM DESASSEMBLEE et COMMENTEE sont quelques-uns des thèmes de cet ouvrage de 700 pages.
Réf. : ML 122
Prix : **249** FF

les plus de Micro Application



LE LANGAGE MACHINE DE L'AMSTRAD CPC (Tome 7)

Ce livre est destiné à tous ceux qui désirent aller plus loin que le BASIC. Des bases de la programmation en assembleur à l'utilisation des routines système, tout est expliqué avec de nombreux exemples. Contient un programme assembleur, moniteur et désassembleur.
Réf. : ML 123
Prix : **129** FF



GRAPHISMES ET SONS DU CPC (Tome 8)

L'AMSTRAD CPC dispose de capacités graphiques et sonores exceptionnelles. Ce livre en montre l'utilisation à l'aide de nombreux programmes utilitaires.
Réf. : ML 124
Prix : **129** FF

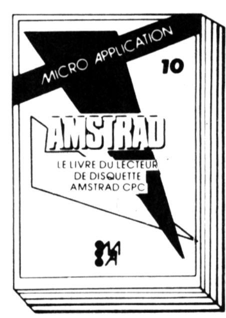
PEEKS ET POKES DU CPC (Tome 9)

Comment exploiter à fond son CPC à partir du BASIC? C'est ce que vous révèle ce livre avec tout ce qu'il faut savoir sur les peeks, pokes et autres call... Vous saurez aussi comment protéger la mémoire, calculer en binaire... et tout cela très facilement. Un passage assuré et sans douleur du BASIC au puissant LANGAGE MACHINE.
Réf. : ML 126
Prix : **99** FF

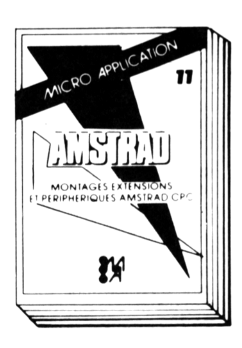


LIVRE DU LECTEUR DE DISQUETTE AMSTRAD CPC (Tome 10)

Tout sur la programmation et la gestion des données avec le 6128 DDI-1 et le 664! Utile au débutant comme au programmeur en langage machine. Contient le listing du DOS commenté, un utilitaire qui ajoute les fichiers RELATIFS a l'AMDOS avec de nouvelles commandes BASIC, un MONITEUR disque et beaucoup d'autres programmes et astuces...

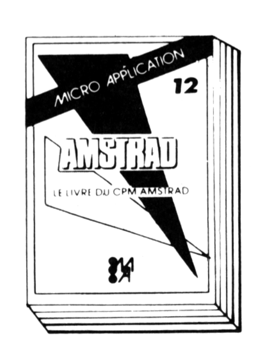


Réf. : ML 127
Prix : **149** FF



MONTAGES, EXTENSIONS ET PERIPHERIQUES AMSTRAD CPC (Tome 11)

Pour tous les amateurs d'électronique, ce livre montre ce que l'on peut réaliser avec un CPC. De nombreux schémas et exemples illustrent les thèmes et applications abordés comme les interfaces, programmateur d'EPROM... Un très beau livre de 450 pages.
Réf. : ML 131
Prix : **199** FF



LE LIVRE DU CP/M AMSTRAD (Tome 12)

Ce livre vous permettra d'utiliser CP/M sur les CPC 464, 664 et 6128 sans aucune difficulté. Vous y trouverez de nombreuses explications et les différents exemples vous assureront une maitrise parfaite de ce très puissant système d'exploitation qu'est CP/M.
Réf. : ML 128
Prix : **149** FF

les livres Amstrad



DES IDÉES POUR LES CPC
(Tome 13)

Vous n'avez pas d'idées pour utiliser votre CPC (464, 664, 6128)? Ce livre va vous en donner! Vous trouverez de très nombreux programmes BASIC couvrant des sujets très variés qui transformeront votre CPC en un bon petit génie. De plus les programmes vous permettront d'approfondir vos connaissances en programmation.

Réf. : ML 132
Prix : **129** FF



LES ROUTINES DE L'AMSTRAD CPC
(Tome 14)

Pour bien connaitre et utiliser les routines utiles de l'AMSTRAD 6128, 664, 464. A la portée de tous. Nombreux programmes utilitaires, exemples, desassembleur, etc.

Ref : ML 143
Prix : **149** FF

DÉBUTER AVEC LE CPC 6128
(tome 15)

Ce livre s'adresse a ceux qui débutent avec le CPC 6128. Tout leur est clairement expliqué aussi bien pour le matériel que pour le logiciel. Une fois leur machine bien en main, ils pourront s'attaquer au Basic et utiliser l'utilitaire de gestion d'adresses que contient le livre.

Réf. : ML 145
Prix : **99** F TTC



LA BIBLE DES CPC 664/6128
(tome 16)

Un régal pour tous ceux qui veulent tout connaitre sur les CPC 6128 et 664. Analyse du système d'exploitation, du processeur, le GATE ARRAY, le contrôleur video, le 8255, le chip sonore, les interfaces... Comprend un desassembleur, les points d'entrée des routines commentés de l'interpréteur et du système d'exploitation. Un super livre comme toutes les Bibles !

Réf. : ML 146 Prix : **199** F





TRUCS ET ASTUCES II POUR CPC
(tome 17)

Ce livre concerne tous les possesseurs de CPC (464, 664 et bien sûr 6128 !). Vous y trouverez un générateur de menus, un générateur de masques, des aides à la programmation comme un DUMP, l'utilisation des routines systèmes et plein d'astuces de programmation. Pour tous ceux qui veulent tirer le maximum de leur CPC !

Réf. : ML 147
Prix : **129** F TTC

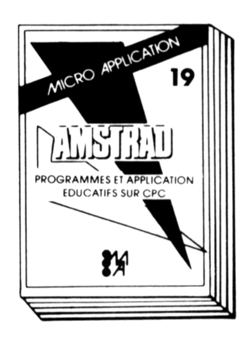


Réf. : ML 148
Prix : **149** FF
Disponible en Mai

LE LIVRE DE LA CAO
(Tome 18)

Avec cet ouvrage vous saurez tout sur la Conception Assistée par Ordinateur et sur la programmation des GRAPHIQUES en 3 dimensions sur les CPC. Les points, lignes, rectangles, cercles, courbes, figures en 3D (comme les cubes, pyramides, cylindres, etc.), les rotations, les effets miroirs, les éclatements et explosions, et enfin pour conclure le clou : toutes les astuces pour créer son propre système de CAO. Nombreux programmes exemples et utilitaires.

les plus de Micro Application



PROGRAMMES et APPLICATIONS ÉDUCATIFS sur CPC. (Tome 19)

Ce livre est un recueil complet de programmes complets et d'applications prêts à fonctionner sur CPC. Chaque programme est très bien commenté et l'ouvrage couvre de nombreux sujets (mathématiques, chimie...). Ce livre est tout particulièrement destiné aux lycéens.

Réf : ML 150
Prix : **179** FF



Disponible en Mai

SYSTÈMES DE TRANSMISSION SUR CPC. (Tome 20)

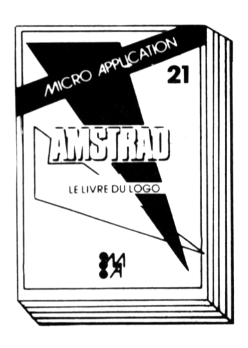
Encore une exclusivité Micro Application. Grâce à ce livre les communications et transmissions n'auront plus de secrets pour vous et vous pourrez profiter au maximum des possibilités offertes aujourd'hui dans ce domaine. Complet avec beaucoup d'applications pratiques, un ouvrage pratique et original.

Réf : ML 151
Prix : **199** F

LE LIVRE DU LOGO (Tome 21)

Le LOGO est un langage très intéressant dont les applications sont très nombreuses. Cet ouvrage permettra au lecteur de profiter au maximum du LOGO livré avec L'AMSTRAD. Principaux thèmes abordés : les graphismes, les procédures, les récursions, les routines de tri, un générateur de masque, structure des données, intelligence artificielle...

Réf. : ML 162
Prix : **149** FF
Disponible en Juin



INTELLIGENCE ARTIFICIELLE ET ROBOTIQUE SUR CPC (Tome 22)

Ce livre est une excellente introduction au monde de l'intelligence artificielle et à ses applications. Toutes les techniques et méthodes décrites sont illustrées de programmes exemples. On apprendra ainsi quelle méthode un robot utilise pour trouver la sortie d'un labyrinthe ou comment un ordinateur peut acquérir des connaissances et ainsi aider à la résolution de problèmes.



Réf. : ML 163
Prix : **149** FF
Disponible en Juin



BIEN DEBUTER AVEC LE PCW

Le premier livre pour l'AMSTRAD PCW! Cet ouvrage vous permettra de réussir à coup sûr vos débuts sur le PCW. On découvre pas à pas le puissant taitement de texte LOCOSCRIPT, puis la programmation BASIC MALLARD et l'utilisation de CP/M. Indispensable pour bien profiter de son PCW.

Réf. : ML 164
Prix : **129** FF



Réf. : ML 165
Prix : **179** FF
Disponible en Juin

LE LIVRE DE L'AMSTRAD PCW

Vous possédez un PCW et vous voulez en tirer le maximum? Alors ce livre a été écrit pour vous! Grâce à lui vous utiliserez au mieux le LOCOSCRIPT et profiterez de toutes les possibilités offertes par le CP/M. Une formation intensive au BASIC MALLARD vous permettra d'écrire des routines d'édition, un générateur de masques de saisie, des routines de tri et une gestion de fichier.


MICRO
INFO
T23
X82
ATARI
AMSTRAD
COMMODORE
COMMODORE
80 pages de trucs et astuces,
programmes, dossiers
pour Amstrad CPC,
Commodore
Atari ST ...
20 f

Achevé d'imprimer en avril 1986
sur les presses de l'imprimerie Laballery
58500 Clamecy
Dépôt légal : avril 1986
Numéro d'imprimeur : 604008

IMPRIMÉ EN FRANCE

[redacted]

Ce livre est un recueil de programmes complets et d'applications éducatifs prêts à fonctionner sur AMSTRAD CPC. Chaque programme est très bien commenté et l'ouvrage couvre de nombreux domaines (mathématiques, langues, chimie, géographie...). Ce livre s'adresse bien sûr en priorité à tous ceux qui souhaitent utiliser leur CPC pour les aider dans leurs études, mais aussi à tous ceux qui désirent élargir leurs connaissances et apprécier les vertus éducatives des AMSTRAD. Enfin la qualité des programmes et leurs commentaires et analyses assureront au lecteur une excellente formation au BASIC.

Aperçu du contenu du livre :

- La programmation en BASIC
- Les Mathématiques :
- Théorème de Pythagore, Equation du second degré
- PGCD et PPCM, etc.
- La Chimie :
- Equation d'une réaction chimique
- La Physique :
- La loi d'Ohm, le principe d'ARCHIMEDE
- Parallélogramme des forces...
- Les Langues :
- Les verbes irréguliers anglais, test de vocabulaire
- La Biologie
- L'Histoire et la Géographie :
- Dates Historiques
- Départements et préfectures
- L'Economie
- Encore des Mathématiques :
- Droite, Cercle, Sinusoïde, Ellipse...

Prix : 179 FF TTC
ISBN : 2-86899-048-7
Réf. : ML 150



AMSTRAD
PROGRAMMES
ET APPLICATIONS
ÉDUCATIFS SUR CPC
19



Document numérisé avec amour par :

AMSTRAD CPC

MÉMOIRE ÉCRITE

https://acpc.me/